Iterative algorithms for total variation-like reconstructions in seismic tomography

Ignace Loris and Caroline Verhoeven
Université Libre de Bruxelles, Brussels, Belgium.

February 27, 2013

Abstract

A qualitative comparison of total variation like penalties (total variation, Huber variant of total variation, total generalized variation, ...) is made in the context of global seismic tomography. Both penalized and constrained formulations of seismic recovery problems are treated. A number of simple iterative recovery algorithms applicable to these problems are described. The convergence speed of these algorithms is compared numerically in this setting. For the constrained formulation a new algorithm is proposed and its convergence is proven.

1 Introduction

The aim of this paper is two-fold. To give a qualitative description of various total variation-like regularization methods in the context of a global seismic inversion problem, and to compare several iterative algorithms numerically, that can be used to perform these inversions. A new algorithm is also included and a proof of convergence is given.

Inverse problems in seismic tomography are characterized by a combination of insufficient and noisy data. It follows that resulting tomographic reconstructions of wave-speed anomalies are non-unique and corrupted by noise; some kind of regularization of the inverse problem is required. The dynamics of the Earth's mantle leads to smooth variations (as a result of heat diffusion) as well as steep gradients (related to the exponential dependence of viscosity on temperature), and sharp transitions (resulting from chemical and mineralogical variations caused by lithosphere subduction and from phase transitions). An important challenge is to retain as much of this in the reconstruction despite the need to regularize. The attraction of simple smoothing algorithms giving simple images, is off-set by smoothing away sharp boundaries.

Here we focus on regularization techniques that have the ability to reconstruct sharp edges. Included in our discussion are the total variation prior (TV) of Rudin et al. [1992] and a number of generalizations and variations on this theme. Among them are the so-called Huber variant of the total variation prior (HTV), the total generalized variation method (TGV) of Bredies et al. [2010], etc. We also include a regularization method that uses a sparse expansion in terms of wavelet coefficients. The methods are described in explicit detail and some typical features of resulting reconstructions are discussed qualitatively in Section 2.

Although details differ, the common theme of these penalties is that they all impose sparsity of certain local differences of the reconstructed model, which is achieved in practice by using a non-smooth convex ℓ_1-norm penalty of these local differences. The effect of using an ℓ_1-norm instead of an ℓ_2-norm squared is that in practice small coefficients are penalized disproportionately more than large coefficients, which leads to sparse reconstructions (i.e. reconstructions with few nonzero coefficients).

The use of an ℓ_1-norm penalty implies that non-linear equations have to be solved in the inversion. We therefore devote a section of the paper to the numerical comparison of the convergence speed of some iterative reconstruction algorithms for these problems. Indeed, a second common feature of these regularization methods is that they can be solved with very similar algorithms (with minor changes).

In this paper we will assume a linear relationship between an unknown model u and data y characterized by a matrix K. One may think e.g. of the matrix K as containing in its rows the discretized versions (on some grid) of ray or finite frequency sensitivity kernels. The central mathematical theme of this paper is then the numerical minimization of some penalized least squares functionals of type:

$$\min_u \frac{1}{2}\|Ku - y\|^2 + \text{penalty}. \tag{1}$$

The first term in this functional represents a quadratic data misfit term that depends on the data y and the matrix K: $\|Kx - y\|^2 = \sum_i (Kx - y)_i^2$. The second term depends on u and serves to regularize the inversion, i.e. it serves to produce a unique model u that satisfies some (qualitative) assumptions imposed on the reconstruction. Here we are principally interested in penalizing local differences Au of the model u in such a way that edges in the model aren't blurred too much. To this end, we will study the problem:

$$\hat{u}(\lambda) = \arg\min_u \frac{1}{2}\|Ku - y\|^2 + \lambda\|Au\|_1. \tag{2}$$

that contains a non-smooth convex ℓ_1-norm penalty of local differences Au of u. The precise choice of the differencing matrix A and the precise form of the ℓ_1-norm $\|\cdot\|_1$ depend on the penalty that is preferred and on the particularities of the model (2D, 3D etc,...). Fully worked out examples of this kind of penalty, which include the TV penalty and generalizations, are presented in Section 2, with further details for solving the associated problem (2) in Sections 3 and 4.

Alternatively, instead of penalizing a least squares term, one may try to solve the constrained problem:

$$\tilde{u}(\epsilon) = \arg\min_{\|Ku-y\|\leq\epsilon} \|Au\|_1. \tag{3}$$

The problems (2) and (3) are equivalent in the sense that for corresponding λ and ϵ, the models $\hat{u}(\lambda)$ and $\tilde{u}(\epsilon)$ are equal; see e.g. [Hennenfent et al., 2008, van den Berg and Friedlander, 2011] for the case $A = 1$. The formulation (3) is useful as the parameter ϵ (related in practice to the amount of noise on the data y) is perhaps easier to estimate than the penalty parameter λ in (2).

In Section 3 we write explicit formulas for several iterative algorithms for the penalized problem (2) and we compare the speed of convergence (in the special case of the TV penalty). The comparison is done for a matrix K that has no special structure (not the identity matrix, not convolution, ...). We also discuss two iterative algorithms for the constrained problem (3). One we believe is new, and we prove convergence. We also make a numerical comparison of the speed of convergence in the constrained case.

Except for one, the algorithms discussed in this paper are fully explicit. They require only application of matrix vector products (with matrices A and K), and one or two simple convex projections. It is important to remark that only the precise form of these projections and the choice of the local differencing operator A differ between the various penalization methods discussed in this paper. Again fully worked out examples are given for the penalties discussed in Section 2.

In section 2 we will review a number of non smooth regularization terms in the context of a toy synthetic seismic tomography experiment. We discuss the effect the various penalty

choices have on the reconstruction. Section 3 lists a number of iterative algorithms that can be used to perform these inversions. A numerical convergence speed comparison is made, both in the penalized case (2) as well as in the constrained case (3). Section 4 explains in detail how these algorithms can be used to solve the problem introduced in Section 2, i.e. we give explicit expressions for the convex projection operators used by the various penalties. Finally, Section 5 contains a proof of convergence of a new iterative minimization algorithm for the constrained problem (3).

2 Comparison of edge-preserving regularization methods

Natural images are often characterized by the presence of both sharp edges and smooth transitions. It is therefore not surprising that much research is done for finding methods of denoising or deconvolving images that can preserve sharp edges. A popular technique is the use of the total variation prior [Rudin et al., 1992], and many generalizations have been proposed. In this section we review a number of these TV-like techniques, compare synthetic 2D reconstructions of a toy model in global seismic tomography and describe their principal features. The description of computational algorithms for these tasks is deferred to Section 3 and specific implementation details for each of the cases discussed are given in Section 4.

2.1 Total variation penalty and generalizations

The total variation (TV) penalty is defined by the ℓ_1-norm of the gradient of the model u, i.e. we choose $A = \text{grad}$ so that the penalty in expressions (1) and (2) becomes:

$$\text{penalty} = \lambda \|Au\|_1 = \lambda \|\text{grad}(u)\|_1 = \lambda \sum_{\text{pixels}} \sqrt{(\Delta_x u)^2 + (\Delta_y u)^2}. \tag{4}$$

Here $\text{grad}(u) = (\Delta_x u, \Delta_y u)$ and $\Delta_x u$ and $\Delta_y u$ are first order local differences of the 2D model u. In formula (4) the sum ranges over all pixels of the model u. This type of regularization will force the gradient of the model to be (exactly) zero in many places: TV promotes sparsity of the gradient of u. In other words, it will give rise to a piece-wise constant reconstruction whenever the data allows for it. This property is lost when the square root in expression (4) is removed. This penalty is translation invariant and isotropic (invariant under rotations). Sometimes a non-isotropic TV is used which is defined as a sum over all pixels of $|\Delta_x u| + |\Delta_y u|$. Formula (4) is easily adapted to 3D models too.

A slight generalization of the TV penalty is the Huber total variation penalty (HTV). Here we replace the $\sqrt{(\Delta_x u)^2 + (\Delta_y u)^2}$ terms of (4) by:

$$\text{penalty} = \lambda \sum_{\text{pixels}} h\left(\sqrt{(\Delta_x u)^2 + (\Delta_y u)^2}\right) \tag{5}$$

where the function h is defined by

$$h(t) = \begin{cases} |t|^2/2\alpha & \text{if} \quad |t| \leq \alpha \\ |t| - \alpha/2 & \text{if} \quad |t| \geq \alpha \end{cases} \tag{6}$$

(see [Huber, 1964, Section 4, point (iii)]). Clearly HTV reduces to TV for $\alpha = 0$. On the other hand, when α is positive, large gradients are penalized as in TV (up to an irrelevant constant), but small gradients are penalized quadratically so that the sparse gradient promoting property of the TV penalty is lost. In practice, local differences will be kept small by this penalty (if possible), but non will be exactly zero as in TV.

A third kind of regularization method that we will consider is a (non-symmetric) total generalized variation penalty (TGV) of [Bredies et al., 2010, remark 3.10]. It also involves an auxiliary variable $v = (v_x, v_y)$ and the problem (2) takes the form

$$\begin{aligned}&\min_{x,v} \frac{1}{2}\|Ku - y\|^2 \\ &+\lambda \sum_{\text{pixels}} \sqrt{(\Delta_x u - v_x)^2 + (\Delta_y u - v_y)^2} + \alpha\sqrt{(\Delta_x v_x)^2 + (\Delta_y v_x)^2 + (\Delta_x v_y)^2 + (\Delta_y v_y)^2}\end{aligned} \tag{7}$$

(see Section 4 for more details). This generalization was designed to yield piece-wise smooth models (instead of piece-wise constant models as in standard TV). The penalty depends on an additional parameter α that controls the balance of the first and second term. Many other generalizations of the TV penalty exist; see e.g. [Chambolle and Lions, 1997, Bredies et al., 2010, Chan et al., 2000].

A straightforward variation on the TV penalty is the use of the ℓ_1-norm, not of the gradient of the model, but of second order differences of the model (i.e. of the local Hessian matrix; $A = \text{Hess}$). We will call it the Hessian penalty (HP):

$$\begin{aligned}\text{penalty} &= \lambda\|Au\|_1 = \lambda\|\text{Hess}(u)\|_1 \\ &= \lambda \sum_{\text{pixels}} \sqrt{(\Delta_x^2 u)^2 + (\Delta_x\Delta_y u)^2 + (\Delta_y\Delta_x u)^2 + (\Delta_y^2 u)^2}\end{aligned} \tag{8}$$

(in each pixel $\text{Hess}(u)$ is a 2×2 matrix; see again Section 4 for details). The use of penalty (8) will force the local Hessian of the model to be zero in most places, i.e. the model will be piecewise linear. Here, any matrix norm on the local Hessian can be chosen; those matrix norms that are expressed in terms of the eigenvalues will yield a penalty that is isotropic. We choose the Frobenius norm as it is easy to work with and isotropic (an explicit expression is given in Section 4).

Regularization strategies that try to express the model as a sparse linear combination of wavelet basis functions [Mallat, 2009] also fit in the category of penalties that use the ℓ_1-norm of local differences. In that case one has

$$\text{penalty} = \lambda\|Au\|_1 = \lambda\|Wu\|_1 \tag{9}$$

were W represents a wavelet transform. A wavelet transform W interleaves local differencing (and averaging) with subsampling. Such a penalty depends directly on the choice of wavelet basis (choice of W). On a cartesian grid, many wavelet families exist [Mallat, 2009] and they are relatively easy to implement (although not as easy as the local differencing used in TV and its variations). Examples include non-smooth wavelet functions such as the simple Haar wavelets, smooth orthogonal wavelets or smooth symmetric wavelets. However, wavelets are non-stationary (preferred positions exist) and 2D and 3D wavelets are non-isotropic (preferred directions exist). Partial solutions to these two problems include the use of the undecimated WT or of directional transforms such as curvelets [Candes et al., 2006] or shearlets [Labate et al., 2005]. Such a strategy has been used in geosciences in [Loris et al., 2007, Hennenfent et al., 2008, Herrmann and Hennenfent, 2008, Gholami and Siahkoohi, 2010, Simons et al., 2011] etc.

2.2 Qualitative comparison on a synthetic example

In the remaining part of this section we apply the various regularization methods described above to a toy problem in seismic tomography. We consider a simple synthetic $2D$ input model u^{input} defined on the globe (see Figure 1, panels a–e). The input model is chosen to have a number of zones of constant value (+1 in blue color or −1 in red color) with either a sharp edge or a

smooth transition in between. Sharp edges are found near North-America. Smooth transitions are found around the Indian Ocean. The edges and transitions are circle-shaped so as not to give preference to any particular direction (i.e. edges are not aligned with the parametrization grid of the sphere). The center of this 'bull's eye' pattern is located at $(36°\mathrm{N}, -120°\mathrm{E})$. The model has circular symmetry around this point.

In panel (b) of Figure 1, a cross section of this model is shown along a great circle passing through the point $(36°\mathrm{N}, -120°\mathrm{E})$ and through the North Pole. The second row of Figure 1 (panels c and d) depict the length of the local gradient of the input model: $|\mathrm{grad}(u^{\mathrm{input}})|$. The gradient is sparse; nonzero values can only be found near edges and transitions in the model. The 'cubed sphere' [Ronchi et al., 1996] was used as a parametrization of the sphere. In this toy experiment the model space has dimension 98304 ($= 128 \times 128 \times 6$ pixels).

In order to set up a synthetic inverse problem, we use a set of 8490 seismic rays corresponding to actual earthquake positions and seismic stations [Trampert and Woodhouse, 1995, 1996, 2001]. The rays (discretized on the $128 \times 128 \times 6$ grid) make up the rows of a 8490×98304 matrix K. In panel (f) of Figure 1 the sum of all rows of the matrix K is shown. It represents the illumination of the globe by the 8490 ray paths. Most rays are concentrated around the Pacific Ocean. The matrix K does not have any special structure that can be exploited by a minimization algorithm.

Next, artificial data y are constructed using the formula $y = Ku^{\mathrm{input}} + n$, where n is chosen as gaussian noise of magnitude $\|n\| = 0.1 \times \|Ku^{\mathrm{input}}\|$ (in other words 10% gaussian noise is added). Our goal is to use the different regularization techniques described in Section 2.1 to obtain faithful reconstructions of u^{input} (from the knowledge of y and K only), and to compare some of their principal characteristics.

We do not expect perfect reconstruction ($u^{\mathrm{output}} \neq u^{\mathrm{input}}$) because the problem is too underdetermined (only 8490 data for 98304 unknowns), and because the data contain noise. The use of TV style regularization methods is appropriate as the model has several areas of constant model value, together with some edges. However, we expect that the TV penalty will unfortunately also enforce piecewise constant model values near the smooth transitions. We compare with the other regularization methods.

Four different reconstructions are made, corresponding to TV, HTV, TGV and HP penalties as described by formulas (4), (5), (7) and (8) in combination with functional (2). The precise iterative algorithm that is used to solve these four instances of problem (2) is described in Section 3, formula (11). A detailed description is deferred to Section 4. At the moment we limit ourselves to discussing some qualitative differences of the resulting output models. It is important to mention right away that all reconstructed models fit the data equally well: $\|Ku^{\mathrm{output}} - y\| = \|n\|$. The four reconstructions are displayed in Figure 2.

In Figure 2, a map view of the four reconstructed models is displayed on the left hand side and a cross section of the four output models is shown in the right hand side column (for reference also a cross section of the input model is shown in dotted lines). As before the cross sections follow the great circle that passes through the point $(36°\mathrm{N}, -120°\mathrm{E})$ and through the North pole. Note however that the output models have lost (some of) their circular symmetry, and a different cross section (along another great circle through the point $(36°\mathrm{N}, -120°\mathrm{E})$) will yield a slightly different profile.

The piece-wise constant nature of the TV output model is clearly noticeable in panels (a) and (b). There is also some loss of amplitude in certain regions. Distinctively, sharp edges (e.g. near North America) are preserved but smooth transitions of the input model (e.g. near Africa and the Indian Ocean) are replaced by a succession of sharp edges (the so-called staircasing effect). In other words a TV penalty imposes a piecewise constant reconstruction as much as the data allows for it. Such a reconstruction always has sharp edges, but it is not guaranteed that the reconstructed edge will be at exactly the same location as in the input model (the

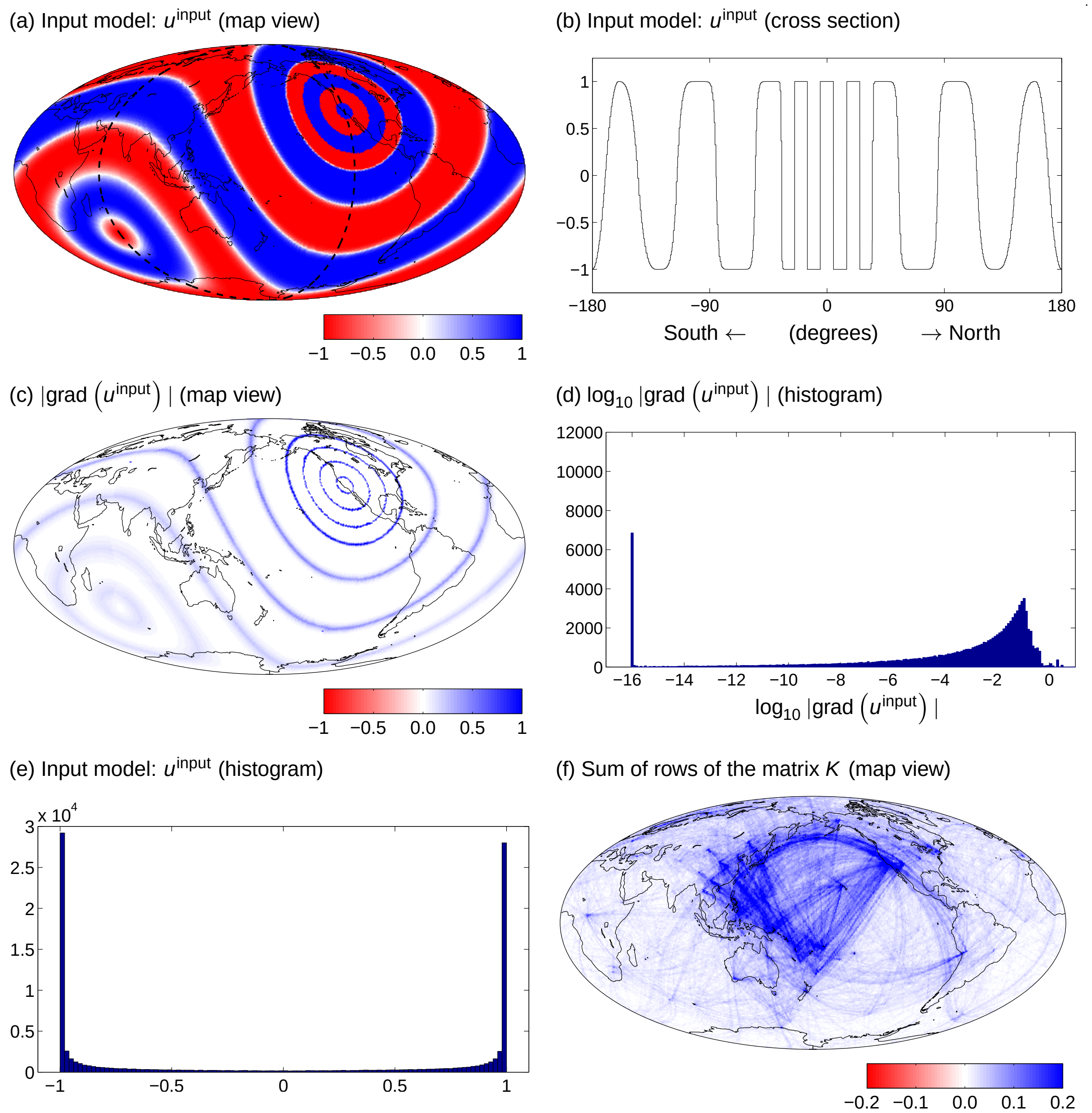

Figure 1: Toy input model for synthetic seismic tomography experiment (see Section 2): (a) input model u^{input} with both sharp and smooth edges between zones of constant model value; (b) cross section along a great circle passing through (36°N, −120°E) (dashed line in panel (a)); the horizontal axis measures the degrees of separation from this point in Northern direction; (c) length of the local gradient of input model (mostly zero except at edges and transitions); (d) histogram of the length of the gradient (on a logarithmic scale; the peak at −16 corresponds zero gradients); (e) Histogram of model values; (f) Illumination of the globe by the 8490 ray paths in the matrix K.

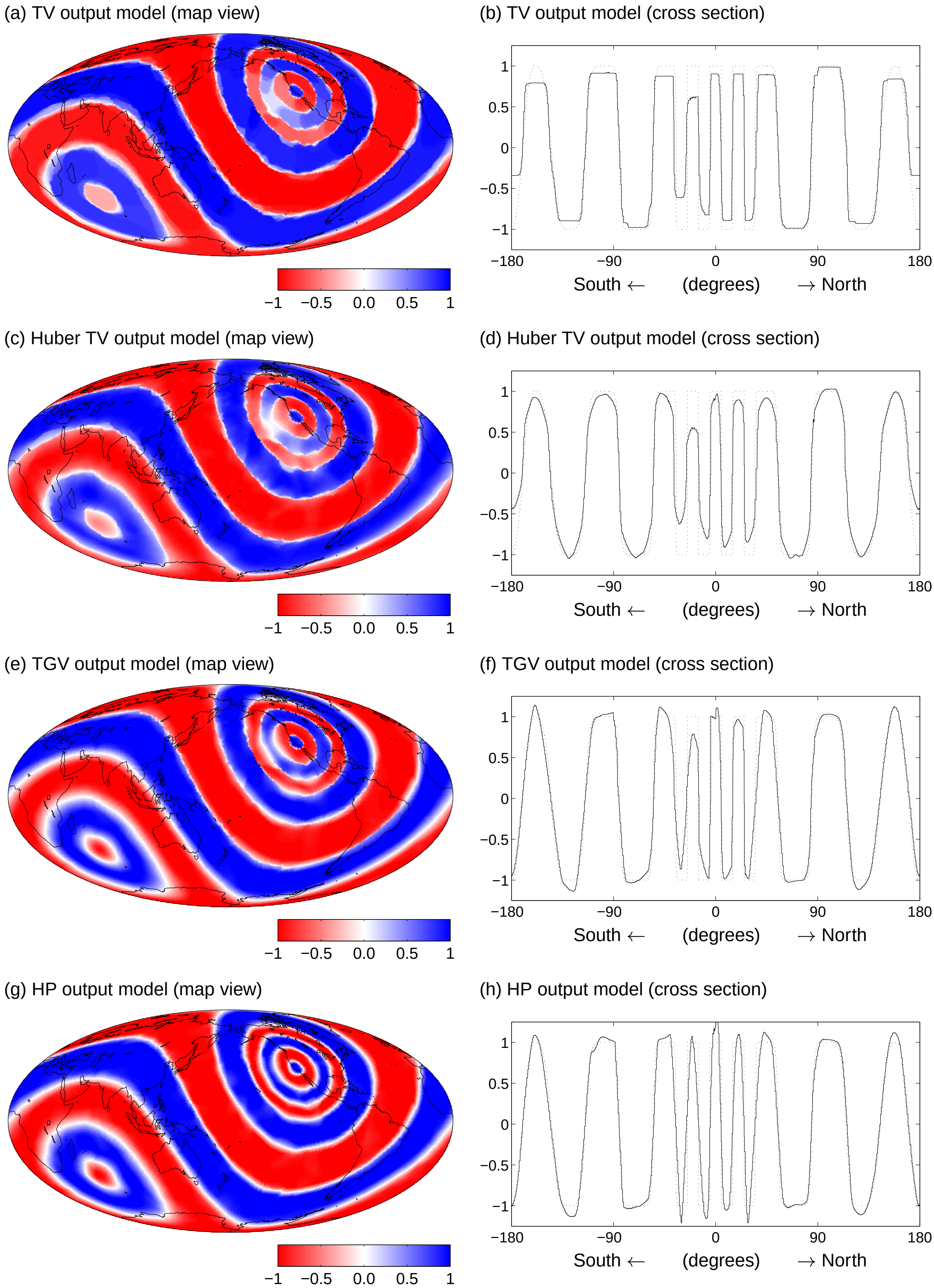

Figure 2: Reconstructions of the input model for four different regularization methods. The cross sections on the right show the output models (solid lines) and the input model (dashed lines). (a)-(b) the total variation reconstruction has an obvious piecewise constant character. The amplitude of the model appears damped in some regions. (c)-(d) the Huber-TV reconstruction is similar to the TV reconstructions but the staircasing effect is less pronounced. (e)-(f) the TGV reconstruction. (g)-(h) the Hessian style reconstruction is piecewise linear.

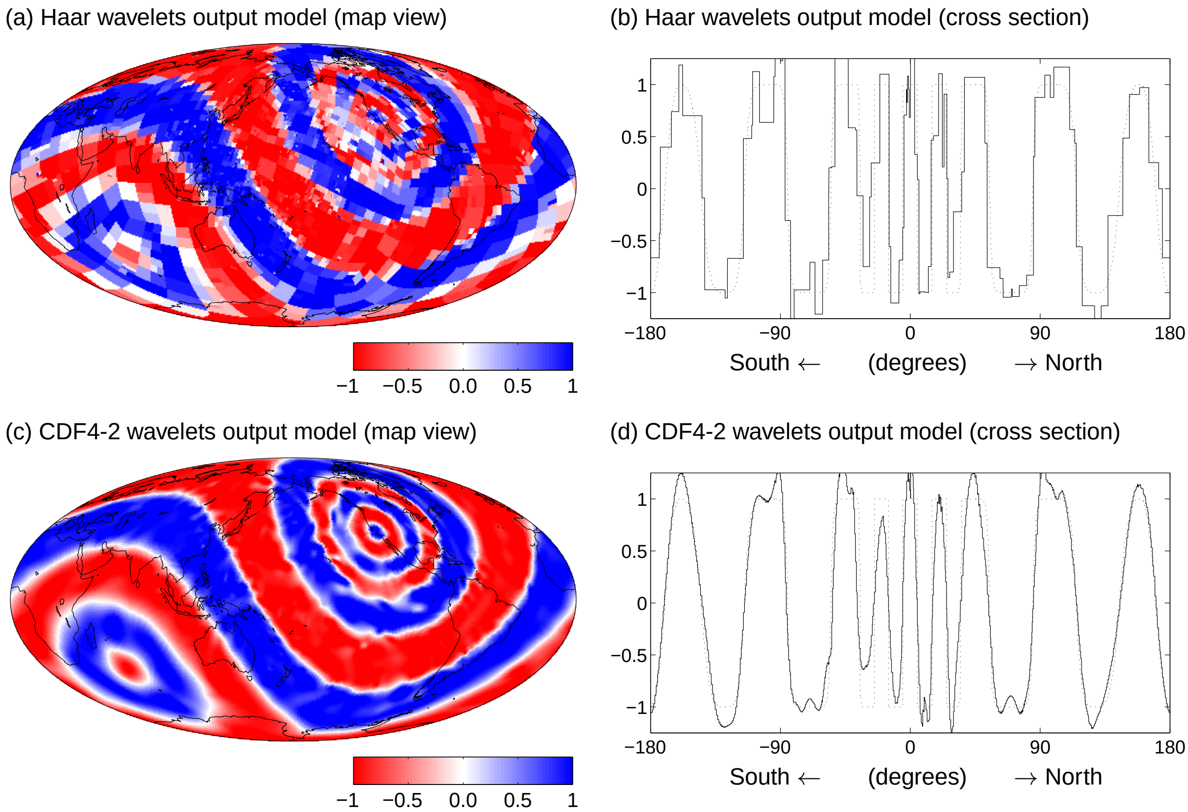

Figure 3: Reconstructions of the input model for two different wavelet regularization methods. The cross sections on the right show the output models (solid lines) and the input model (dashed lines). (a)-(b) a reconstruction that is sparse in the Haar wavelet basis. (c)-(d) a reconstruction that is sparse in the CDF4-2 wavelet basis. The Haar model uses only 1668 nonzero wavelet basis coefficients, and the CDF 4-2 reconstruction uses a mere 996 nonzero wavelet basis coefficients.

position is influenced by noise on the data and by the limited number of rays).

The Huber TV reconstruction looks similar to the TV reconstruction but it has edges that are somewhat smoother. The piecewise constant nature of the reconstruction is lost. It should be said that the precise amount of smoothing depends on the value of the extra parameter α that is present in the Huber penalty. For $\alpha = 0$ one recovers the TV reconstruction, but for large α one obtains a reconstruction that is penalized by the ℓ_2-norm squared of the gradient.

The TGV reconstruction exhibits edges that are comparable to the Huber TV reconstructions, but amplitudes are damped less. Some edges are replaced by linear transitions. The TGV penalty also depends on a parameter α that can be tuned between the TV case (large α) and the HP case (small α). Linear transitions also characterize the HP reconstruction. Here the penalty is proportional to the 1-norm of the second derivative of the model. For the example shown, the difference between the TGV reconstruction lies somewhere between the TV and HP reconstruction.

Finally, Figure 3 shows two reconstructions that were obtained by imposing a ℓ_1-norm penalty of wavelet coefficients of the model. Panels (a) and (b) of this Figure show a reconstruction that is sparse in the Haar wavelet basis. The number of nonzero basis coefficients is 1668 (out of a possible 98304). These wavelets are orthonormal, but non-smooth. The second example (panels c and d) uses a wavelet family with very smooth wavelets (CDF 4-2) of Cohen et al. [1992]. This model has only 996 nonzero coefficients (out of a possible 98304). The smoothness of the wavelet bases functions is clearly reflected in the nature of the reconstructions shown in Figure 3. These two reconstruction fit the 8490 data as well as the four previous reconstructions. The Haar wavelet reconstruction is not visually appealing. Even though the Haar basis functions are piece-wise constant, the Haar reconstruction does not look similar to the TV reconstruction. If wavelets are used, one not only needs to choose the wavelet family (Haar, ...), but also the number of levels of the wavelet transform. In this example, we chose to make a 4 level wavelet transform.

The original input model possesses an azimuthal symmetry around the point (36°N, −120°E). However, the reconstructed models have lost this symmetry. This is due to the lack of data, the uneven distribution of the ray coverage, the noise on the data and the penalties imposed. For instance, the (circular) edges that are present in the input model are reconstructed, but the precise position is changed depending on this azimuthal angle. Instead of showing a single cross section of the output models, as was done in the second column of Figures 2 and 3, it makes sense to also calculate the cross sections along many great circles passing through the same point (36°N, −120°E) and average them. The average over 36 great circles is displayed in Figure 4; in the case of the TV reconstruction one sees that such an average cross section does not exhibit the same sharp edges as a single cross section does. The cause is simply that the output model is not circularly symmetric around this point. In other words the reconstructed edges do not necessarily coincide with the original edges. Averaging this effect leads to some smoothing. This also occurs in the Haar reconstruction. The effect is less evident on the other reconstruction that are already smooth. Figure 5 shows a map view of the difference $|u^{\text{output}} - u^{\text{input}}|$ between input model and reconstruction.

Figure 6 shows the length of the local gradient of four reconstructed models in map view and as a histogram. These should be compared to Figure 1, panels (c) and (d). The gradient of the TV reconstructed model in panels (a) and (b) is sparse, although $|\text{grad}(u^{\text{out}})|$ is nowhere exactly equal to zero. This is a result of stopping the iterative reconstruction algorithm after a finite number of iterations (in this case after 1000 iterations). This behavior is best seen on the histogram where the peak between 10^{-4} and 10^{-6} represents values of $|\text{grad}(u^{\text{out}})|$ that have not fully converged to zero. A black line shows the position of the histogram after 10^5 iterations. We can clearly see that the secondary peak has moved to the left, i.e. to values around 10^{-8}. The primary peak stays roughly in the same position, indicating little change to the significant

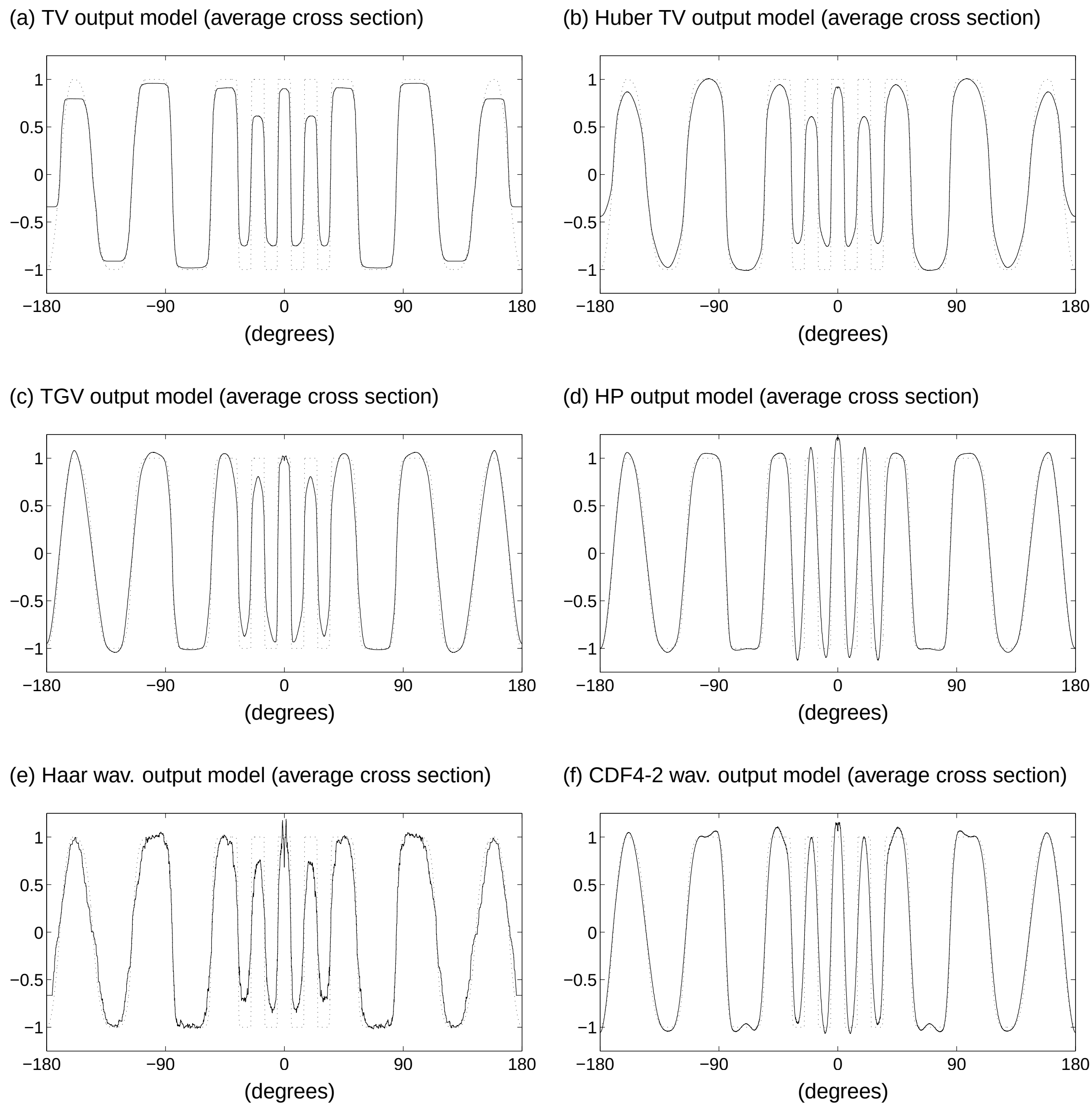

Figure 4: Averaged cross sections of the six reconstructions. The average is taken over 36 great circles passing through the point (36°N, −120°E), and with azimuth $0, 10, 20, \ldots, 350$ degrees w.r.t local North.

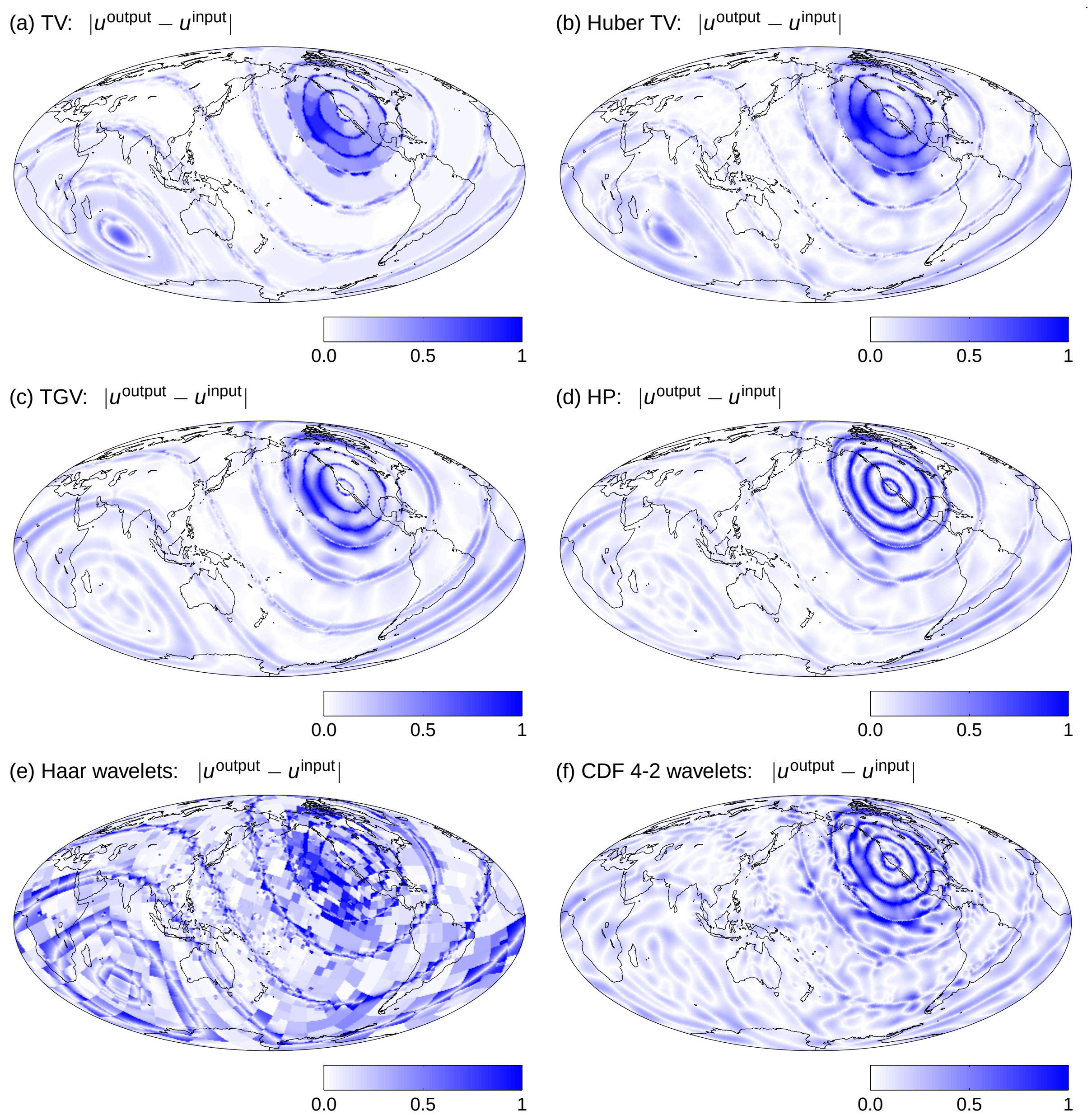

Figure 5: Map view of the differences $|u^{\text{output}} - u^{\text{input}}|$ for the six reconstructions. Large errors may occur near edges (as a result of smoothing or change of edge position in u^{output}) and in other places (as a result of amplitude damping in u^{output}).

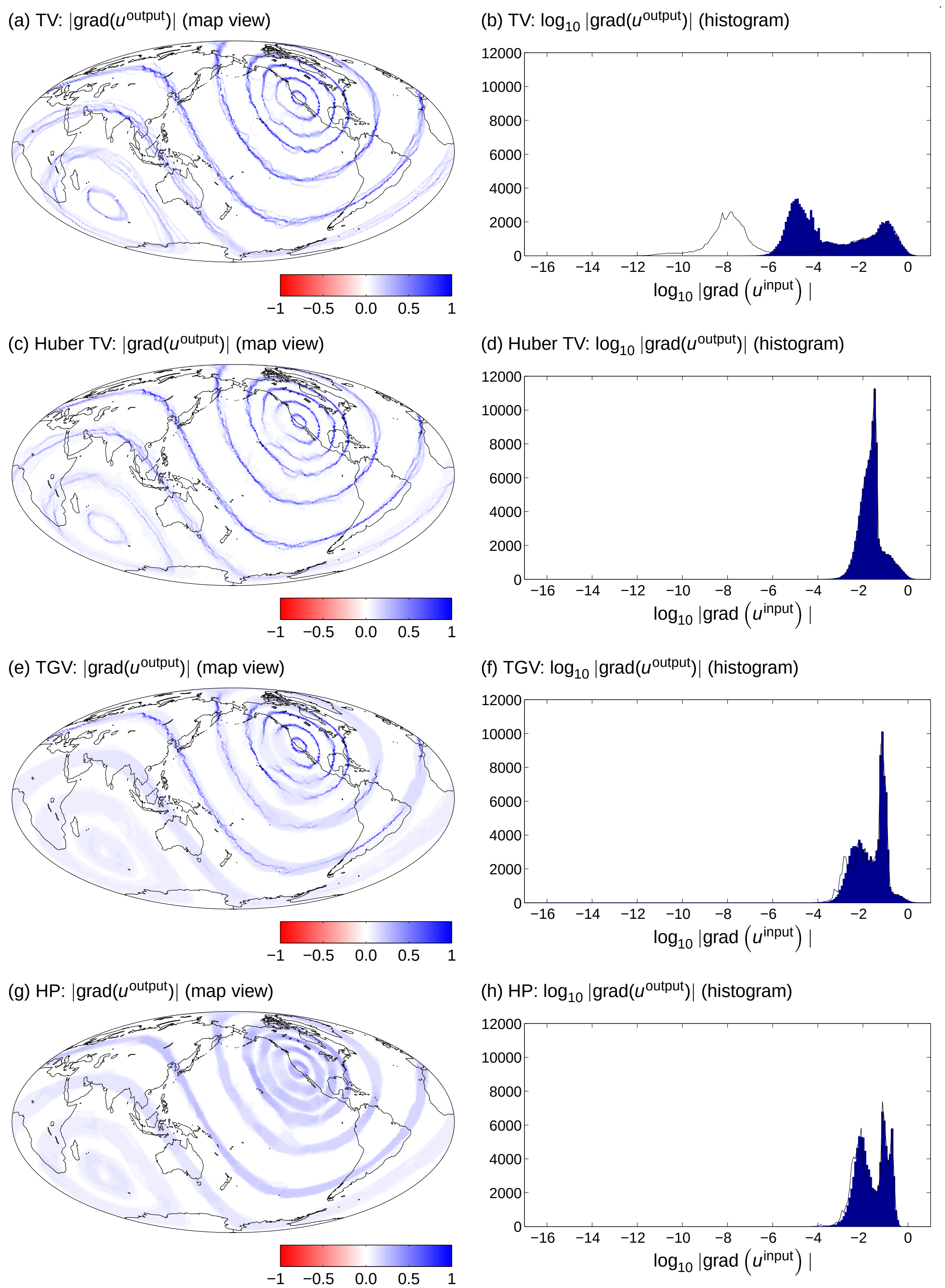

Figure 6: Gradient field of four reconstructed models of Section 2. On the left hand side is a map view of the norm of the local gradient of the output models. On the right hand side is a histogram (of the logarithm) of the norm of the local gradient of these output models (in blue, after 1000 iterations; a black line indicates the position of the histogram after 10^5 iterations). For the first reconstruction (TV), one expects this gradient field to be zero in most places. The Huber-TV reconstruction penalizes the local gradient but small (nonzero) values remain. For the last two reconstructions one does not expect sparse (or almost sparse) gradients, but a piecewise constant gradient.

values of the gradient after 1000 iterations.

On the second row of Figure 6 we see that the Huber TV penalty does not lead to sparse gradients (for $\alpha \neq 0$). Although many values are small (colored in white on the map view), there is no heavy tail visible on the left hand side of the corresponding histogram in panel (d).

The TGV and HP penalties (panels (e)–(h)) give rise to local gradients that are markedly different in character from the TV and Huber TV cases. Here, as in the Huber TV case, we do not expect sparse gradients. The HP reconstruction imposes a piece-wise linear solution, as much as the data allows for. Therefore the gradient of the output model will be piecewise constant. This is clearly observed in panel (e) of Figure 6. In case of the TGV reconstruction, the results lie somewhere between the TV reconstruction and the HP reconstruction.

In case of the HP reconstruction, we expect a model that has sparse Hessian field (if one would plot the Frobenius norm of the local Hessian (see formulas (8), (28) and (30) for explicit expressions) the sparse nature of this Hessian field would be apparent). In other words, the second derivatives of the model will be mostly zero. This means that the output model will be piecewise linear.

The six imaging models (4)–(9) give rise to qualitatively different reconstructions. The associated minimization problems (2) or (3) however, can all be solved using the same iterative algorithms, with minor variations. The next section contains a description and numerical comparison of several suitable algorithms. Section 4 contains the technical details on how these algorithms can be applied to the penalized problems discussed above.

3 Algorithms

In this section we review a number of iterative algorithms that can be used for solving problems (2) and (3) and make a numerical comparison of them. We start with some known algorithms for the penalized problem (2) in Subsection 3.1.

In Subsection 3.2 we perform the same kind of comparison for two iterative algorithms for the constrained problem (3). One is a primal dual hybrid gradient algorithm of Esser et al. [2010] and the second one is, as far as the authors know, new. A proof of convergence of the second algorithm is therefore included in Section 5.

The algorithms presented generally involve matrix-vector multiplications and vector space operations (addition, multiplication with scalar). Furthermore they also involve a simple convex projection operator P_λ or a simple soft-thresholding operation S_λ. These are defined component-wise by $P_\lambda((w_1,\dots,w_N)) = (P_\lambda(w_1),\dots,P_\lambda(w_N))$ and $S_\lambda((w_1,\dots,w_N)) = (S_\lambda(w_1),\dots,S_\lambda(w_N))$ with:

$$P_\lambda(w_i) = \begin{cases} \frac{w_i}{\|w_i\|}\lambda & \|w_i\| > \lambda \\ w_i & \|w_i\| \leq \lambda \end{cases} \qquad \text{and} \qquad S_\lambda(w_i) = w_i - P_\lambda(w_i). \tag{10}$$

Here w_i may itself be an element of $\mathbb{R}, \mathbb{R}^2, \dots$. We refer to Section 4 for some more information. The precise details may vary depending on wether one treats a 2D or 3D problem, or the precise form of the penalty (TV, Huber TV etc).

Below we will use the symbol $\|A\|^2$ to denote the largest eigenvalue of $A^T A$, and $\|K\|^2$ to denote the largest eigenvalue of $K^T K$. Apart from the model variable u, the iterative algorithms below also use one or several auxiliary variables $\bar{u}$, w (which is the subgradient of $\lambda\|\cdot\|_1$), etc. The starting point of each of the algorithms in Subsections 3.1 and 3.2 is arbitrary.

3.1 Algorithms for penalized problems

The generalized iterative soft-thresholding algorithm [Loris and Verhoeven, 2011]:

$$\left\{\begin{array}{lcl} \bar{u}^{n+1} & = & u^n + \tau_1 K^T(y - Ku^n) - \tau_1 A^T w^n \\ w^{n+1} & = & P_\lambda\left(w^n + \frac{\tau_2}{\tau_1} A\bar{u}^{n+1}\right) \\ u^{n+1} & = & u^n + \tau_1 K^T(y - Ku^n) - \tau_1 A^T w^{n+1}, \end{array}\right. \tag{11}$$

converges to a minimizer of the penalized problem (2) if step sizes τ_1 and τ_2 are chosen as $\frac{1}{2}\tau_1\|K\|^2 < 1$ and $\tau_2\|A\|^2 < 1$.

When applied to problem (2) the algorithm of Chambolle and Pock [2011] takes the form:

$$\left\{\begin{array}{lcl} w^{n+1} & = & P_\lambda\left(w^n + \sigma\frac{\tau_2}{\tau_1} A\bar{u}^n\right) \\ v^{n+1} & = & \frac{1}{1+\sigma}v^n + \frac{\sigma}{1+\sigma}(K\bar{u}^n - y) \\ u^{n+1} & = & u^n - \tau_1 K^T v^{n+1} - \tau_1 A^T w^{n+1} \\ \bar{u}^{n+1} & = & u^{n+1} + \theta\left(u^{n+1} - u^n\right). \end{array}\right. \tag{12}$$

It converges to a minimizer of problem (2) for $\theta = 1$ and $\sigma\|\tau_1 K^T K + \tau_2 A^T A\| < 1$.

An explicit Bregman algorithm:

$$\left\{\begin{array}{lcl} u^{n+1} & = & u^n + \tau_1 K^T(y - Ku^n) - \tau_1 A^T\left[w^n + \frac{1}{\theta}(w^n - w^{n-1})\right] \\ w^{n+1} & = & (1-\theta)w_n + \theta P_\lambda\left(w^n + \frac{\tau_2}{\tau_1} A u^{n+1}\right) \end{array}\right. \tag{13}$$

converges for $0 < \theta \leq 1$ and $\|\tau_1 K^T K + \tau_2 A^T A\| < 1$ to a minimizer of problem (2). It is found from [Zhang et al., 2011, Equations 5.9, 5.10 and 5.11].

The conditions for convergence on K and A for algorithms (12) and (13) are different from the condition in algorithm (11). In the last two algorithms K and A are coupled in a single condition, whereas two separate conditions are used for algorithm (11). In other words, the conditions in algorithm (11) only depend on the matrix norms of K and A, whereas in algorithms (12) and (13) the conditions also depend on the orientation of the singular vectors of K with respect to the singular vectors of A. This makes algorithm (11) a little bit easier to use. Also, no $\frac{1}{2}$ is found in front of $K^T K$in the conditions for algorithms (12) and (13).

It is also worth noting that for $\sigma = 1$ and $\theta = 1$ the algorithm (12) reduces to algorithm (13) for $\theta = 1$. We omit the details of this calculation.

The above three algorithms are fully explicit: they only require the application of the matrix K and its transpose at every step, and the application of the operator A and its transpose. The nonlinear operator P_λ also has a simple implementation (see also Section 4 for explicit expressions).

In addition to the three explicit algorithms above, we also include the following implicit algorithm in our comparison (here implicit means that a linear system needs to be solved in every step):

$$\left\{\begin{array}{lcl} u^{n+1} & = & (K^T K + \alpha A^T A)^{-1}\left(K^T y - A^T(w^n - \alpha z^n)\right) \\ z^{n+1} & = & S_{\lambda/\alpha}\left(Au^{n+1} + w^n/\alpha\right) \\ w^{n+1} & = & w^n + \alpha(Au^{n+1} - z^{n+1}) \end{array}\right. \tag{14}$$

for $0 < \theta \leq 1$ and a parameter α. It is the so-called split-Bregman method [Goldstein and Osher, 2009].

The split Bregman method was used in a seismic tomography context in [Gholami and Siahkoohi, 2010]. The method seems appropriate for some special matrices K and A, for which the inverse $(K^T K + \alpha A^T A)^{-1}$ is easy (such as e.g. the combination convolution K and A =grad which can both be diagonalised in Fourier space) or when legacy code exists: If K and A have no special structure, the first line in the split-Bregman algorithm (14) itself needs an iterative algorithm.

The above four algorithms are compared in the framework of the TV reconstruction of Section 2, i.e. using the matrix K and the synthetic data y of Secion 2 and the choice A =grad. A reference minimizer u^{ref} of problem (2) was first obtained using 10^5 iterations of algorithm (11) with $\tau_1 = 1.99/\|K\|^2$ and $\tau_2 = 0.99/\|A\|^2$. This was done for two values of λ. One value of λ yields a reconstructed model that fits the data to noise level $\|Ku^{\mathrm{ref}} - y\|/\|n\| \approx 1$, and another (larger) value of λ that underfits the data: $\|Ku^{\mathrm{ref}} - y\|/\|n\| \approx 3$. In this way, it is possible to evaluate the performance of the algorithms for different values of λ. A larger value of λ would be used for cases with more noise; a smaller value of λ would be used for cases with less noise.

In Figure 7, panels (a) and (b), the evolution of the relative error $\|u^n - u^{\mathrm{ref}}\|/\|u^{\mathrm{ref}}\|$ to the true minimizer is plotted as a function of computing time for these four algorithms. The algorithms ran for 1000 iterations each (starting from zero model), except the split-Bregman algorithm which ran for only 200 outer iterations (and 5 inner iterations of conjugate gradient type). On the horizontal axis computing time is used rather than number of iterations as the split-Bregman algorithm's outer iteration step is more expensive than a single step of the explicit algorithms. Indeed, the explicit algorithms use a single application of the matrices K, K^T, A and A^T in each iteration whereas the implicit split-Bregman algorithm needs to solve a linear system in each step. The times mentioned in Figure 7 were obtained for this specific example on a single 2.66GHz CPU with 12GB memory running Matlab R2009a. They cannot easily be extrapolated to other problems (with different numbers of variables, data, other matrices K and A, different hardware). Panels (c) and (d) show the evolution of the functional (2) as a function of time. For algorithms (11) and (12), one can show that the functional (2) tends to its limiting value as $\mathcal{O}(1/n)$ where n is the number of iterations [Loris and Verhoeven, 2011, Chambolle and Pock, 2011]. There is no such result for the error $\|u^n - \hat{u}\|$.

The algorithms that perform the best for the smaller value of λ perform the worst for the larger value of λ. Moreover we see that the smallest value of $\|u^n - u^{\mathrm{ref}}\|/\|u^{\mathrm{ref}}\|$ does not necessarily correspond to the smallest value of $F(u^n) - F(u^{\mathrm{ref}})$. We conclude that all algorithms mentioned are suitable for solving problem (2), but that convergence may depend on a good choice of the step size parameters used. For algorithm (11) the larger step size $\tau_1 = 1.99/\|K\|^2$ is suitable for cases with low noise, whereas the choice $\tau_1 = 0.99/\|K\|^2$ is better in high noise cases. Generally, the error between u^n and the true minimizer lies between 1 and 10% after 1000 iterations.

It was already remarked that these algorithms do not give sparse Au^n at every iteration step; only for the limiting value of u^∞ will Au^∞ be sparse. When A is the unit matrix or an orthogonal matrix, then the generalized iterative soft threshdolding algorithm (11) reduces to the traditional soft-thresholding algorithm of Daubechies et al. [2004] which does produce sparse u^n at every step:

$$u^{n+1} = S_{\lambda\tau_1}\left(u^n + \tau_1 K^T(y - Ku^n)\right) \quad \text{for} \quad \min_u \frac{1}{2}\|Ku - y\|^2 + \lambda\|u\|_1 \tag{15}$$

for step sizes $\tau_1\|K\|^2 < 2$. Such a simplified algorithm could be used for sparse recovery in a wavelet basis as was done in [Loris et al., 2007] in a seismic tomography context. It suffices to make the change of variables $u = W^{-1}w$ in expressions (2) and (9) to use algorithm (15) with K replaced by KW^{-1} and u^n replaced by w^n. Such a change of variables is not possible for the TV penalty as A = grad is not invertible. An accelerated (more efficient) version of algorithm (15), the so-called Fast Iterative Soft-Thresholding Algorithm (FISTA), is described in [Beck and Teboulle, 2009] (for $\tau_1\|K\|^2 < 1$). See also Yamagishi and Yamada [2011] for some recent developments.

For $K = 1$ (or orthogonal) and $\tau_1 = 1$ the algorithm (11) reduces to the following projected

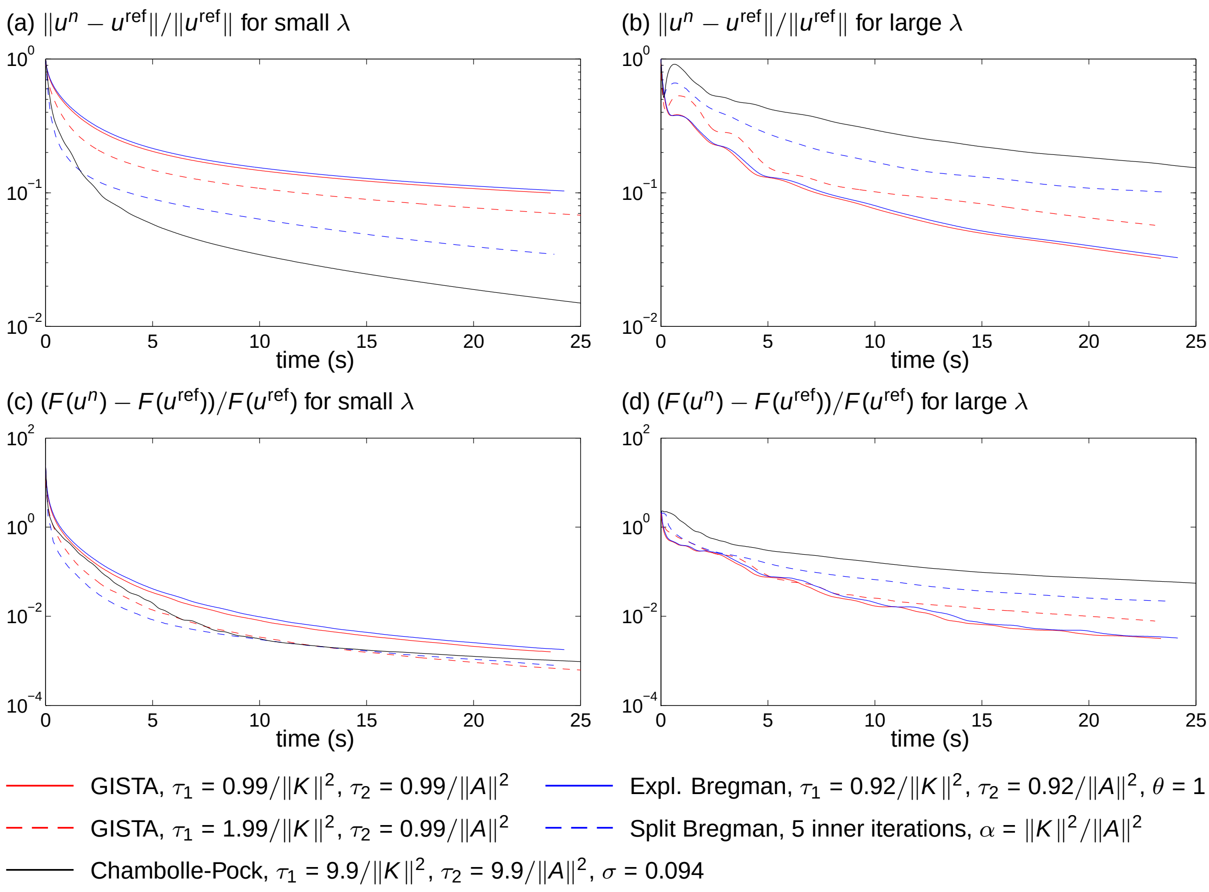

Figure 7: Convergence rate of the iterative minimization algorithms of Section 3.1, with relative error to the true minimizer on the top and value of the functional on the bottom. The right hand side column refers to an experiment with a larger penalty parameter and left hand side column refers to an experiment with a smaller penalty parameter. The best algorithm depends on the value of the penalty parameter λ, on the step sizes used, and on the criteria (distance to true minimizer or value of functional). Computing time instead of number of iterations is used for fair comparison.

gradient algorithm:

$$\begin{cases} w^{n+1} &= P_\lambda\left(w^n + \tau_2 A(K^T y - A^T w^n)\right) \\ u^{n+1} &= K^T y - A^T w^{n+1} \end{cases} \quad \text{for} \quad \min_u \frac{1}{2}\|u-y\|^2 + \lambda\|Au\|_1 \tag{16}$$

for step size $\tau_2\|A\|^2 < 2$ [Chambolle, 2005]. This algorithm can also be accelerated [Nesterov, 1983]. The case $K = 1$ and $A = \mathrm{grad}$ corresponds to denoising with a total variation penalty. A recent comprehensive comparison of various numerical algorithms for this task can be found in Section 6.2.1 of [Chambolle and Pock, 2011].

3.2 Algorithms for constrained problems

In this subsection we compare two explicit iterative algorithms for the constrained minimization problem (3). This problem depends on the parameter ϵ which determines the desired data misfit: $\|Kx-y\| \le \epsilon$. Such a formulation of a reconstruction problem is therefore useful when one wants to fit the data to noise level. It then suffices to set the parameter ϵ equal to the norm of the noise vector (Morozov discrepancy principle).

In addition to the projection operator P_λ the constrained algorithms below also use another projection operator $Q_{y,\epsilon}$ defined as:

$$Q_{y,\epsilon}(v) = \begin{cases} y + \epsilon\frac{v-y}{\|v-y\|} & \text{for} \quad \|v-y\| > \epsilon \\ v & \text{for} \quad \|v-y\| \le \epsilon \end{cases} \quad \text{and} \quad T_{y,\epsilon}(v) = v - Q_{y,\epsilon}(v) \tag{17}$$

given y and ϵ. In other words, $Q_{y,\epsilon}$ is the projection on the ℓ_2 ball of radius ϵ centered at y and $T_{y,\epsilon}$ is an associated thresholding function.

We start with the so-called primal-dual hybrid gradient algorithm (PDHGMp) found in equation 5.3 of [Esser et al., 2010]. In the present notation this algorithm takes the form:

$$\begin{cases} u^{n+1} &= u^n - \tau_1 K^T(v^n + (v^n - v^{n-1})/\theta) - \tau_1 A^T(w^n + (w^n - w^{n-1})/\theta) \\ w^{n+1} &= P_{\mu/\tau_1}\left(w^n + \frac{\tau_2}{\tau_1} A u^{n+1}\right) \\ v^{n+1} &= (1-\theta)v^n + \theta T_{y,\epsilon}\left(v^n + K u^{n+1}\right) \end{cases} \tag{18}$$

and converges for $0 < \theta \le 1$, $\mu > 0$, $\|\tau_1 K^T K + \tau_2 A^T A\| < 1$. This algorithm can also be derived from algorithm (A_1) on page 28 of [Zhang et al., 2011] and (for $\theta = 1$) from algorithm 1 in [Chambolle and Pock, 2011]. We omit the details.

Another algorithm is given by the formulas:

$$\begin{cases} \bar{u}^{n+1} &= u^n - \tau_1 K^T(v^n + (v^n - v^{n-1})/\theta) - \tau_1 A^T w^n \\ w^{n+1} &= P_{\mu/\tau_1}\left(w^n + \frac{\tau_2}{\tau_1} A \bar{u}^{n+1}\right) \\ u^{n+1} &= u^n - \tau_1 K^T(v^n + (v^n - v^{n-1})/\theta) - \tau_1 A^T w^{n+1} \\ v^{n+1} &= (1-\theta)v^n + \theta T_{y,\epsilon}\left(v^n + K u^{n+1}\right) \end{cases} \tag{19}$$

and it is proven in Section 5 that it converges to the minimizer of problem (3) for $0 < \theta \le 1$, $\mu > 0$, $\tau_1\|K\|^2 < 1$ and $\tau_2\|A\|^2 < 1$. We will refer to this algorithm as the generalized basis pursuit denoising algorithm (GBPDNA). The motivation for this name is given at the end of this Section.

In Figure 8 the two algorithms are compared numerically on the same problem as in Subsection 3.1 (i.e. same K, A and y). Moreover, we chose two values of ϵ corresponding to the value of $\|Ku^{\mathrm{ref}} - y\|$ taken by the two reference minimizers of the simulation in the Subsection 3.1. In other words, we chose ϵ so that the corresponding minimizers are identical to the minimizers of the numerical simulation in Subsection 3.1. Because of this choice of ϵ it is possible to directly compare panels (a) and (b) of Figure 8 with panels (a) and (b) of Figure 7.

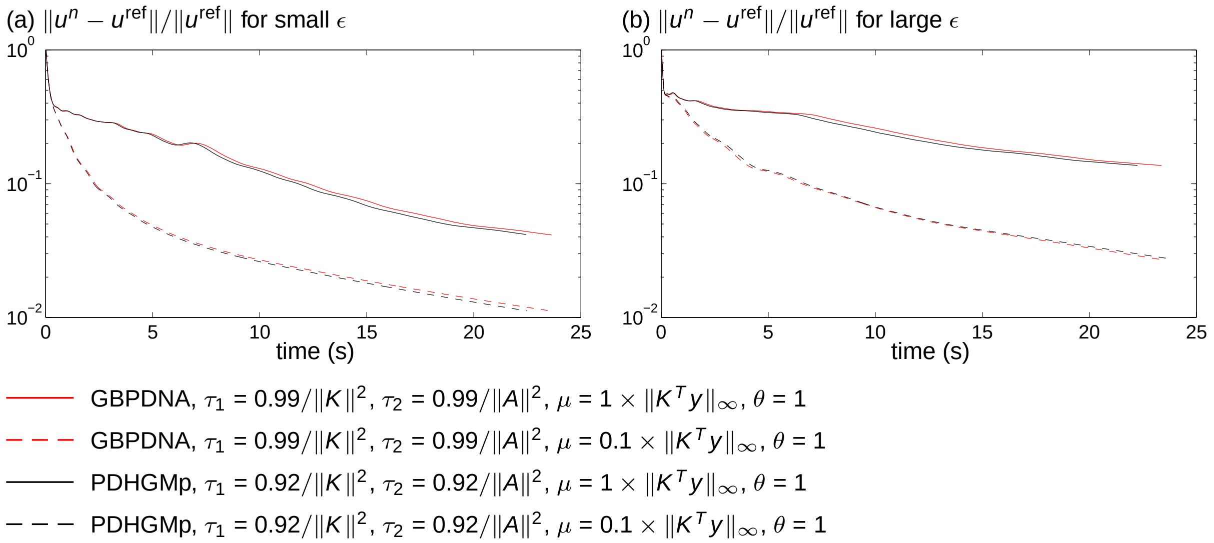

Figure 8: Convergence rate of two iterative minimization algorithms for the constrained problem (3). Pictured is the relative error to the true minimizer (obtained from algorithm (19) and 10^5 iterations). The figure on the left hand side is for a small value of ϵ and the one on the right is for a larger value of ϵ. Each algorithm was run for 1000 iterations.

We recompute the true (reference) minimizer u^{ref} with 10^5 iterations of algorithm (19) with $\tau_1 = 0.99/\|K\|^2$, $\tau_2 = 0.99/\|A\|^2$ and $\mu = \|K^T y\|_\infty$. We verified that the resulting minimizer is equal (up to 0.1%) to the reference minimizer of subsection 3.1. Then we compare the iterates of algorithms (18) and (19) to these reference minimizers. We clearly notice that both algorithms are almost identical in their convergence behavior. The effect of an appropriate choice of the parameter μ in these algorithms is also visible in Figure 8. We conclude that there is no significant difference in convergence speed between these two algorithms (18) and (19) for the constrained minimization problem (3). In practice it is slightly easier to work with two conditions of type $\tau_1\|K\|^2 < 1$ and $\tau_2\|A\|^2 < 1$ than one condition of type $\|\tau_1 K^T K + \tau_2 A^T A\| < 1$.

The constrained problem (3) reduces to the so-called 'basis pursuit denoising' problem:

$$\arg \min_{\|Ku-y\| \leq \epsilon} \|u\|_1 \tag{20}$$

when $A = 1$, and to 'basis pursuit' [Chen et al., 1998]

$$\arg \min_{Ku=y} \|u\|_1 \tag{21}$$

when $A = 1$ and $\epsilon = 0$. In the case of problem (20) the proposed algorithm (19) reduces to:

$$\left\{ \begin{array}{lcl} u^{n+1} & = & S_\mu(u^n - \tau_1 K^T(v^n + (v^n - v^{n-1})/\theta)) \\ v^{n+1} & = & (1-\theta)v^n + \theta T_{y,\epsilon}\left(v^n + Ku^{n+1}\right) \end{array} \right. \tag{22}$$

and in case of problem (21) the proposed algorithm (19) reduces to

$$\left\{ \begin{array}{lcl} u^{n+1} & = & S_\mu(u^n - \tau_1 K^T(v^n + (v^n - v^{n-1})/\theta)) \\ v^{n+1} & = & (1-\theta)v^n + \theta(v^n + Ku^{n+1} - y). \end{array} \right. \tag{23}$$

The last special case (23) is the same as algorithm 5.6 of [Zhang et al., 2011]. For this reason we will call algorithm (19) the generalized basis pursuit denoising algorithm (GBPDNA).

In algorithms (22) and (23) the step size parameter τ_1 satisfies $\tau_1\|K\|^2 < 1$. As a result of the soft-thresholding, the u^n are sparse in every step.

4 Explicit formulas for the algorithms used in Section 2

The iterative algorithms of Sections 3.1 and 3.2 were compared in the framework of TV -penalized seismic recovery (equations (2) and (4)). The same algorithms, with minor modifications, can also be used to solve the other three problems of Section 2 as well. In this section we give some explicit formulas for the expressions encountered in these penalties and algorithms. The details depend e.g. on the number of spatial dimensions (2D, or 3D), on the precise expression of the differencing matrix A in functionals (2) and (3), etc.

For the TV regularized problem (4) we set $A = \mathrm{grad}$, which we can define as $\mathrm{grad}(u) = (\Delta_x u, \Delta_y u)$ and :

$$(\Delta_x u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & i: 1 \dots N-1 \\ 0 & i = N \end{cases} \qquad (\Delta_y u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & j: 1 \dots N-1 \\ 0 & j = N \end{cases} \tag{24}$$

$(i, j : 1 \dots N)$ for $u \in \mathbb{R}^{N\times N}$. So for a 2D model $u \in \mathbb{R}^{N\times N}$, the gradient field $\mathrm{grad}(u)$ will be in $\mathbb{R}^{N\times N\times 2}$. The transpose of A is then given by the formulas $A^T(w_x, w_y) = \Delta_x^T w_x + \Delta_y^T w_y$ with:

$$\begin{aligned} &(\Delta_x^T w_x)_{i,j} = \begin{cases} -w_{x,i,j} + w_{x,i-1,j} & i: 2 \dots N-1 \\ -w_{x,i,j} & i = 1 \\ w_{x,i-1,j} & i = N \end{cases} \\ &\text{and} \\ &(\Delta_y^T w_y)_{i,j} = \begin{cases} -w_{y,i,j} + w_{y,i,j-1} & j: 2 \dots N-1 \\ -w_{y,i,j} & j = 1 \\ w_{y,i,j-1} & j = N \end{cases} \end{aligned} \tag{25}$$

$(i, j : 1 \dots N)$ for a $w_x, w_y \in \mathbb{R}^{N\times N}$. With these definitions one has that $\langle Au, w\rangle = \langle u, A^T w\rangle$ for all $u \in \mathbb{R}^{N\times N}$ and all $w \in \mathbb{R}^{N\times N\times 2}$.

These operations are easy to code (e.g. in MATLAB) and the extension of the above formulas to 3D models ($u \in \mathbb{R}^{N\times N\times N}$) is straightforward. In the examples of Section 2 we used the cubed sphere parametrization of Ronchi et al. [1996]. The formulas for grad (or Δ_x and Δ_y) and grad^T (or Δ_x^T and Δ_y^T) are the same as in (24) and (25), except that other boundary conditions are used to ensure the correct behavior at the edges of the six ('square') faces that make up the parametrization of the sphere.

For minimizing the functional (4) the generalized iterative soft-thresholding algorithm (11) was used. The nonlinear operator P_λ appearing in it, in this case becomes in accordance with formula (10):

$$P_\lambda(w_x, w_y) = \begin{cases} \dfrac{\lambda}{\sqrt{w_x^2 + w_y^2}}(w_x, w_y) & \sqrt{w_x^2 + w_y^2} > \lambda \\ (w_x, w_y) & \sqrt{w_x^2 + w_y^2} \le \lambda, \end{cases} \tag{26}$$

for $(w_x, w_y) \in \mathbb{R}^2$. Here we have dropped the double subscript $i, j \in \{1 \dots N\}$ for clarity.

For the Huber TV regularization method, one uses the same expressions for $A = \mathrm{grad}$ and A^T as in the TV case, but the operator P_λ has to be replaced by:

$$P_{\lambda,\alpha}(w_x, w_y) = \begin{cases} \dfrac{\lambda}{\sqrt{w_x^2 + w_y^2}}(w_x, w_y) & \sqrt{w_x^2 + w_y^2} > \lambda + \alpha \\ \dfrac{\lambda}{\lambda+\alpha}(w_x, w_y) & \sqrt{w_x^2 + w_y^2} \le \lambda + \alpha, \end{cases} \tag{27}$$

again applied in every pixel. With this minor modification the algorithms of Section 3 may also be applied to problem (5). The convergence is still guaranteed under the same conditions as in Theorem 1 (see also note after proof in Section 5).

For the problem (8), we have set $Au = \mathrm{Hess}(u)$ with

$$\mathrm{Hess}(u) = \begin{pmatrix} \Delta_x^2 u & \Delta_x \Delta_y u \\ \Delta_y \Delta_x u & \Delta_y^2 u \end{pmatrix} \tag{28}$$

for $u \in \mathbb{R}^{N \times N}$, i.e. $\mathrm{Hess}(u) \in \mathbb{R}^{N \times N \times 2 \times 2}$. Its transpose A^T is given by the formula:

$$\mathrm{Hess}^T(w) = (\Delta_x^T)^2 w_{11} + \Delta_y^T \Delta_x^T w_{12} + \Delta_x^T \Delta_y^T w_{21} + (\Delta_y^T)^2 w_{22} \tag{29}$$

acting on a $w \in \mathbb{R}^{N \times N \times 2 \times 2}$, with Δ_x^T and Δ_y^T defined as in (25). The Frobenius norm of the local Hessian, as used in penalty (8), is:

$$\left\| \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \right\|_F = \sqrt{H_{11}^2 + H_{12}^2 + H_{21}^2 + H_{22}^2} \tag{30}$$

for $H \in \mathbb{R}^{2 \times 2}$. It is equal to $\sqrt{\sigma_1^2 + \sigma_2^2}$ where σ_1, σ_2 are the singular values of H. In formula (8) the sum over all pixels of expression (30) is taken, i.e.:

$$\lambda \|Ax\|_1 = \lambda \sum_{\text{pixels}} \left\| \begin{pmatrix} \Delta_x^2 u & \Delta_x \Delta_y u \\ \Delta_y \Delta_x u & \Delta_y^2 u \end{pmatrix} \right\|_F . \tag{31}$$

The advantage of using this matrix norm lies not only in the fact that it is easy to compute but also in the fact that the penalty becomes isotropic. In other words, a rotation of the coordinate axis, does not change this expression (the rotation $(x', y') = R(x, y)$ leads to $H' = RHR^T$ and therefore $\|H'\|_F = \|H\|_F$). Other spectral matrix norms (i.e. norms that are based on the spectrum of H), besides the Frobenius norm, also lead to isotropic penalties. However, the Frobenius norm is particularly easy to use as it does not require an explicit singular value decomposition of each H (in each pixel) to compute it. And, as we shall see, the associated projection P_λ is also easy to compute. Note that e.g. the penalty $\sum_{ij} |(\Delta u)_{ij}|$ is also isotropic. It leads to models where the laplacian Δu of u will be mostly zero, i.e. models that are piecewise harmonic (not piecewise linear).

In the case (8) the expression for P_λ used in algorithm (11) is

$$P_\lambda \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} = \begin{cases} \lambda \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \Big/ \left\| \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \right\|_F & \left\| \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \right\|_F > \lambda \\ \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} & \left\| \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \right\|_F \leq \lambda, \end{cases} \tag{32}$$

for $(w_{11}, w_{12}; w_{21}, w_{22}) \in \mathbb{R}^{2 \times 2}$. Again we have dropped the double subscript i, j for clarity but it is understood that the above formula should be applied in every pixel. As already mentioned, this expression for P_λ does not require a singular value decomposition of H in every pixel.

In case of the total generalized variation penalty (7), one has a functional of the form $\frac{1}{2}\|Ku - y\|^2 + \lambda(\|\mathrm{grad}(u) - v\|_1 + \alpha \|Dv\|_1)$. It can therefore be treated with a combination of the above formulas for $P_\lambda, \mathrm{grad}, \Delta_x, \Delta_y$, etc. The matrix A in algorithm (11) now takes the form $\begin{pmatrix} \mathrm{grad} & -\mathrm{Id} \\ 0 & \alpha D \end{pmatrix}$ and acts on (u, v) where v is an auxiliary variable in $\mathbb{R}^{N \times N \times 2}$. The operator D acting on v is a derivative: $Dv = \begin{pmatrix} \Delta_x v_x & \Delta_y v_x \\ \Delta_x v_y & \Delta_y v_y \end{pmatrix}$. The explicit expression for penalty (7) involves therefore:

$$\sum_{\text{pixels}} \sqrt{(\Delta_x u - v_x)^2 + (\Delta_y u - v_y)^2} + \alpha \sqrt{(\Delta_x v_x)^2 + (\Delta_y v_x)^2 + (\Delta_x v_y)^2 + (\Delta_y v_y)^2}. \tag{33}$$

Name	$\|u^{\text{output}} - u^{\text{input}}\|/\|u^{\text{input}}\|$	$\|Ku^{\text{output}} - y\|/\|n\|$
TV	0.20643	1.0009
Huber-TV	0.22519	1.0041
TGV	0.20843	1.0003
HP	0.20033	1.0080
Haar	0.39789	1.0034
CDF 4-2	0.25248	1.0059

Table 1: Reconstruction results for the six reconstructions of Section 2. All reconstructions fit the data equally well (third column), and have slightly different reconstruction error (second column).

This penalty is denoted by ¬symTGV in [Bredies et al., 2010]. The explicit expression for the nonlinear projection operator P_λ used in the iterative algorithms now combines both expressions (27) (for the first line of A) and (32) (for the second line of A). In this case the norm $\|A\|$ depends on the parameter α.

For the simulation of Section 2, the algorithm (11) was implemented four times with the above definitions of A and P_λ. For those cases, we chose $\tau_1 = 1.99/\|K\|^2$ and $\tau_2 = 0.99/\|A\|^2$ and 1000 iterations were performed. 1000 iterations correspond to about 20 seconds of computer time on a single 2.66GHz CPU with 12GB of memory, running Matlab R2009a. The relative error with the input model, $\|x^{\text{output}} - x^{\text{input}}\|/\|x^{\text{input}}\|$, is given in Table 1 for each of the six reconstructions. Two sparse wavelet reconstruction were also calculated in Section 2. Here 1000 iterations of the FISTA algorithm [Beck and Teboulle, 2009] were used with appropriate wavelet transform W (see formula (9) and algorithm (15) with $K \to KW^{-1}$).

None of the six output models u^{output} is expected to be exactly equal to the input model u^{input} because of the lack of data, the noise on the data and the effect of the penalties on the reconstructions.

The computational complexity of the algorithms depends on the matrices K and A (or W). It was already mentioned in Section 3 that the iterative algorithms discussed in this paper use a single application of K, K^T, A and A^T in every step (except for algorithm (14)). The matrix K encodes the relationship between the model u and data y. In case K is a dense matrix, a single application of K (or its transpose) requires $\mathcal{O}(mN)$ operations. Here N is the number of components of u and m is the number of data. If the matrix K is sparse (as is the case for the examples in this paper), or has structure (as e.g. in deconvolution) then this can be reduced significantly. If the matrix A is chosen as a local differencing operator, then applying A (or its transpose) requires $\mathcal{O}(N)$ operations. The same is true for a wavelet transform W. The convex projections P_λ are also $\mathcal{O}(N)$ operations (as they are applied componentwise). The operator $T_{y,\epsilon}$ used in the constrained algorithms (18) and (19) has computational complexity $\mathcal{O}(m)$. The computing times mentioned in this paper do not serve as a basis for extrapolation to other situations; what is important here is that the application of such A and P_λ typically takes less time than applying K and K^T. This is what makes the application of the edge-preserving penalties of Section 2 possible in practice.

5 Proof of convergence of constrained algorithm (19)

Without loss of generality we set $\tau_1 = \tau_2 = 1$ in algorithm (19) and prove convergence for $\|K\| < 1$ and $\|A\| < 1$ (the step sizes can be introduced by scaling the matrices K, A and the

data y). The algorithm (19) can thus be written as:

$$\left\{\begin{array}{lcl} \bar{u}^{n+1} & = & u^n - K^T(v^n + Ku^n - z^n) - A^T w^n \\ w^{n+1} & = & P_\mu(w^n + A\bar{u}^{n+1}) \\ u^{n+1} & = & u^n - K^T(v^n + Ku^n - z^n) - A^T w^{n+1} \\ z^{n+1} & = & Q_{y,\epsilon}(Ku^{n+1} + v^n) \\ v^{n+1} & = & v^n + \theta(Ku^{n+1} - z^{n+1}). \end{array}\right. \tag{34}$$

with $\|K\| < 1$ and $\|A\| < 1$. Indeed, the variable z^{n+1} can be eliminated from the last line of (34) to yield the last line of (19):

$$\begin{array}{rcl} v^{n+1} & = & v^n + \theta\left(Ku^{n+1} - z^{n+1}\right) \\ & = & (1-\theta)v^n + \theta\left(Ku^{n+1} + v^n - Q_{y,\epsilon}\left(Ku^{n+1} + v^n\right)\right) \\ & = & (1-\theta)v^n + \theta T_{y,\epsilon}\left(Ku^{n+1} + v^n\right) \end{array}$$

where we used $T_{y,\epsilon}(a) = a - Q_{y,\epsilon}(a)$. Substitution of the last line of (34) (with $n \to n-1$) in the first line of algorithm (34) then yields the first line of algorithm (19).

The minimizer of problem (3) is determined by its variational equations. These are derived as follows. Introducing a positive parameter μ, we write problem (3) as $\min_u \mu\|Au\|_1 + I(Ku)$, where I is the indicator function of the ℓ_2 ball of radius ϵ around y: $I(Ku) = 0$ for $\|Ku - y\| \leq \epsilon$ and $I(Ku) = \infty$ for $\|Ku - y\| > \epsilon$. The variational equations of the constrained problem (3) are therefore

$$A^T w + K^T v = 0, \tag{35}$$

with w an element of the subdifferential of $\mu\|\cdot\|_1$ at Au and v an element of the subdifferential of I at Ku.

The subdifferential w of $\mu\|Ax\|_1$ satisfies $w_i = \mu\,(Au)_i/|(Au)_i|$ for $(Au)_i \neq 0$ and $|w_i| \leq \mu$ for $(Au)_i = 0$. This means that $(Au)_i = S_\mu(w_i + (Au)_i)$ or equivalently, using (10), that $w_i = P_\mu(w_i + (Au)_i)$, which we write as $w = P_\mu(w + Au)$.

Similarly one shows that the subdifferential of v of I at Ku is characterized by the relation $Ku = Q_{y,\epsilon}(Ku + v)$. The variational equations that determine the minimizer of problem (3) can therefore be written with an auxiliary variable $z = Ku$ as:

$$\left\{\begin{array}{lcl} w & = & P_\mu\,(w + Au) \\ u & = & u - K^T(v + Ku - z) - A^T w \\ z & = & Q_{y,\epsilon}\,(Ku + v) \\ v & = & v + \theta\,(Ku - z)\,, \end{array}\right. \tag{36}$$

which correspond to the fixed point equations of iteration (34) when $\theta \neq 0$.

Lemma 1. *Let P_C be a projection on a non-empty closed convex set $C \subseteq \mathbb{R}^N$. Let $u^+, u^-, \Delta \in \mathbb{R}^N$. If $u^+ = P_C(u^- + \Delta)$, then*

$$\|u^+ - u\|^2 \leq \|u^- - u\|^2 - \|u^+ - u^-\|^2 - 2\langle u - u^+, \Delta\rangle \tag{37}$$

for all $u \in C$.

Proof. Because P_C is the projection on the convex set C we have:

$$\langle u - P(u'), u' - P_C(u')\rangle \leq 0$$

for all $u \in C$ and all u'. Setting $u' = u^- + \Delta$ and $P_C(u') = u^+$ in this inequality yields

$$\langle u - u^+, u^- + \Delta - u^+\rangle \leq 0.$$

As $\langle u - u^+, u^- - u^+\rangle = \left(\|u - u^+\|^2 + \|u^- - u^+\|^2 - \|u - u^-\|^2\right)/2$ we find the relation (37). □

The proof of the following theorem is a combination of the proof of theorem 2 in [Loris and Verhoeven, 2011] and theorem 4.2 of [Zhang et al., 2011].

Theorem 1. *In a finite dimensional setting, and when $\|K\| < 1$, $\|A\| < 1$ and $0 < \theta \leq 1$, the iteration (34) converges to a fixed point and provides a minimizer of problem (3) .*

Proof. It was already remarked that the fixed point equations (36) correspond to the variational equations for problem (3). The problem (3) has a solution and therefore a solution $(\hat{u}, \hat{w}, \hat{v}, \hat{z})$ to the equations (36) exists.

Taking into account line 2 of algorithm (34) and lemma 1 with $u^+ = w^{n+1}$, $u^- = w^n$, $\Delta = A\bar{u}^{n+1}$, $u = \hat{w}$, and $P_C = P_\mu$, we find that:

$$\|w^{n+1} - \hat{w}\|^2 \leq \|w^n - \hat{w}\|^2 - \|w^{n+1} - w^n\|^2 - 2\langle \hat{w} - w^{n+1}, A\left(u^{n+1} - A^T(w^n - w^{n+1})\right)\rangle$$

where we have also used $\bar{u}^{n+1} = u^{n+1} - A^T(w^n - w^{n+1})$. Similarly one finds from the first line of (36) and application of lemma 1 (with $u^+ = \hat{w}$, $u^- = \hat{w}$, $\Delta = A\hat{u}$, $u = w^{n+1}$ and $P_C = P_\mu$) that:

$$\|\hat{w} - w^{n+1}\|^2 \leq \|\hat{w} - w^{n+1}\|^2 - \|\hat{w} - \hat{w}\|^2 - 2\langle w^{n+1} - \hat{w}, A\hat{u}\rangle.$$

Together these two inequalities yield:

$$\begin{aligned} \|w^{n+1} - \hat{w}\|^2 &\leq \|w^n - \hat{w}\|^2 - \|w^{n+1} - w^n\|^2 \\ &\quad -2\langle \hat{w} - w^{n+1}, A\left((u^{n+1} - \hat{u}) - A^T(w^n - w^{n+1})\right)\rangle \\ &= \|w^n - \hat{w}\|^2 - \|w^{n+1} - w^n\|^2 + \|A^T(w^{n+1} - \hat{w})\|^2 \\ &\quad +\|A^T(w^{n+1} - w^n)\|^2 - \|A^T(w^n - \hat{w})\|^2 \\ &\quad -2\langle \hat{w} - w^{n+1}, A(u^{n+1} - \hat{u})\rangle. \end{aligned} \tag{38}$$

In the same way, the third line of algorithm (34) and lemma 1 (with $u^+ = u^{n+1}$, $u^- = u^n$, $\Delta = -K^T(v^n + Ku^n - z^n) - A^T w^{n+1}$, $u = \hat{u}$, $P_C = \mathrm{Id}$) implies

$$\|u^{n+1} - \hat{u}\|^2 \leq \|u^n - \hat{u}\|^2 - \|u^{n+1} - u^n\|^2 - 2\langle \hat{u} - u^{n+1}, -K^T(v^n + Ku^n - z^n) - A^T w^{n+1}\rangle$$

and the second line of (36) with lemma 1 (with $u^+ = \hat{u}$, $u^- = \hat{u}$, $\Delta = -K^T(\hat{v} + K\hat{u} - \hat{z}) - A^T\hat{w}$, $u = u^{n+1}$, $P_C = \mathrm{Id}$) implies:

$$\|\hat{u} - u^{n+1}\|^2 \leq \|\hat{u} - u^{n+1}\|^2 - \|\hat{u} - \hat{u}\|^2 - 2\langle u^{n+1} - \hat{u}, -K^T(\hat{v} + K\hat{u} - \hat{z}) - A^T\hat{w}\rangle,$$

such that together they yield:

$$\begin{aligned} \|u^{n+1} - \hat{u}\|^2 &\leq \|u^n - \hat{u}\|^2 - \|u^{n+1} - u^n\|^2 \\ &\quad -2\langle \hat{u} - u^{n+1}, -K^T\left(v^n - \hat{v} + K(u^n - \hat{u}) - (z^n - \hat{z})\right) - A^T(w^{n+1} - \hat{w})\rangle \\ &\leq \|u^n - \hat{u}\|^2 - \|u^{n+1} - u^n\|^2 \\ &\qquad +2\langle K(\hat{u} - u^{n+1}), K(u^n - \hat{u})\rangle - 2\langle K(\hat{u} - u^{n+1}), z^n - \hat{z}\rangle \\ &\qquad +2\langle \hat{u} - u^{n+1}, K^T(v^n - \hat{v}) + A^T(w^{n+1} - \hat{w})\rangle \\ &= \|u^n - \hat{u}\|^2 - \|u^{n+1} - u^n\|^2 \\ &\qquad -\|K(u^{n+1} - \hat{u})\|^2 + \|K(u^{n+1} - u^n)\|^2 - \|K(u^n - \hat{u})\|^2 \\ &\qquad +\|K(u^{n+1} - \hat{u})\|^2 - \|\hat{z} - z^n - K(\hat{u} - u^{n+1})\|^2 + \|z^n - \hat{z}\|^2 \\ &\qquad -2\langle \hat{u} - u^{n+1}, -K^T\left(v^n - \hat{v}\right) - A^T(w^{n+1} - \hat{w})\rangle \\ &= \|u^n - \hat{u}\|^2 - \|u^{n+1} - u^n\|^2 + \|K(u^{n+1} - u^n)\|^2 \\ &\qquad -\|K(u^n - \hat{u})\|^2 - \|\hat{z} - z^n - K(\hat{u} - u^{n+1})\|^2 + \|z^n - \hat{z}\|^2 \\ &\qquad -2\langle \hat{u} - u^{n+1}, -K^T\left(v^n - \hat{v}\right) - A^T(w^{n+1} - \hat{w})\rangle. \end{aligned} \tag{39}$$

And from the fourth line of algorithm (34) and lemma 1 (with $u^+ = z^{n+1}$, $u^- = 0$, $\Delta = Ku^{n+1} + v^n$, $u = \hat{z}$ and $P_C = Q_{y,\epsilon}$) one finds:

$$\|z^{n+1} - \hat{z}\|^2 \leq \|0 - \hat{z}\|^2 - \|z^{n+1} - 0\|^2 - 2\langle \hat{z} - z^{n+1}, Ku^{n+1} + v^n\rangle.$$

Similarly, from the third line of (36) and lemma 1 with $u^+ = \hat{z}$, $u^- = 0$, $\Delta = K\hat{u} + \hat{v}$, $u = z^{n+1}$ and $P_C = Q_{y,\epsilon}$ one has:

$$\|z^{n+1} - \hat{z}\|^2 \leq \|0 - z^{n+1}\|^2 - \|\hat{z} - 0\|^2 - 2\langle z^{n+1} - \hat{z}, K\hat{u} + \hat{v}\rangle$$

which together yield:

$$2\|z^{n+1} - \hat{z}\|^2 \leq 2\langle z^{n+1} - \hat{z}, K(u^{n+1} - \hat{u}) + v^n - \hat{v}\rangle$$

or:

$$\begin{array}{rcl} 0 & \leq & 2\langle z^{n+1} - \hat{z}, -(z^{n+1} - \hat{z}) + K(u^{n+1} - \hat{u}) + v^n - \hat{v}\rangle \\ & = & -\|z^{n+1} - \hat{z}\|^2 - \|\hat{z} - z^{n+1} + K(u^{n+1} - \hat{u})\|^2 + \|K(u^{n+1} - \hat{u})\|^2 \\ & & +2\langle z^{n+1} - \hat{z}, v^n - \hat{v}\rangle. \\ & = & -\|z^{n+1} - \hat{z}\|^2 - \theta^{-2}\|v^{n+1} - v^n\|^2 + \|K(u^{n+1} - \hat{u})\|^2 \\ & & +2\langle z^{n+1} - \hat{z}, v^n - \hat{v}\rangle, \end{array} \tag{40}$$

where we have used the last line of (34) and of (36).

Finally from the last line of (34) and from lemma 1 (with $u^+ = v^{n+1}$, $u^- = v^n$, $\Delta = \theta(Ku^{n+1} - z^{n+1})$, $u = \hat{v}$ and $P_C = \mathrm{Id}$) it follows that:

$$\|v^{n+1} - \hat{v}\|^2 \leq \|v^n - \hat{v}\|^2 - \|v^{n+1} - v^n\|^2 - 2\theta\langle \hat{v} - v^{n+1}, Ku^{n+1} - z^{n+1}\rangle.$$

In the same way it follows from the last line of the fixed-point equation (36) and lemma 1 (with $u^+ = \hat{v}$, $u^- = \hat{v}$, $\Delta = \theta(K\hat{u} - \hat{z})$, $u = v^{n+1}$ and $P_C = Id$) that:

$$\|v^{n+1} - \hat{v}\|^2 \leq \|v^{n+1} - \hat{v}\|^2 - \|\hat{v} - \hat{v}\|^2 - 2\theta\langle v^{n+1} - \hat{v}, K\hat{u} - \hat{z}\rangle$$

and together the last two expressions yield:

$$\begin{array}{rcl} \theta^{-1}\|v^{n+1} - \hat{v}\|^2 & = & \theta^{-1}\|v^n - \hat{v}\|^2 - \theta^{-1}\|v^{n+1} - v^n\|^2 \\ & & \quad +2\langle v^{n+1} - \hat{v}, K(u^{n+1} - \hat{u}) - (z^{n+1} - \hat{z})\rangle \\ & = & \theta^{-1}\|v^n - \hat{v}\|^2 - \theta^{-1}\|v^{n+1} - v^n\|^2 \\ & & \quad +2\langle v^{n+1} - v^n, K(u^{n+1} - \hat{u}) - (z^{n+1} - \hat{z})\rangle \\ & & \quad +2\langle v^n - \hat{v}, K(u^{n+1} - \hat{u}) - (z^{n+1} - \hat{z})\rangle \\ & \overset{\text{last lines of (34) and (36)}}{=} & \theta^{-1}\|v^n - \hat{v}\|^2 + \theta^{-1}\|v^{n+1} - v^n\|^2 \\ & & \quad +2\langle v^n - \hat{v}, K(u^{n+1} - \hat{u}) - (z^{n+1} - \hat{z})\rangle. \end{array} \tag{41}$$

Adding inequalities (38), (39), (40) and (41) together yields:

$$\begin{array}{l} \|u^{n+1} - \hat{u}\|^2 + \|w^{n+1} - \hat{w}\|^2 + \theta^{-1}\|v^{n+1} - \hat{v}\|^2 \leq \|u^n - \hat{u}\|^2 - \|u^{n+1} - u^n\|^2 \\ \quad +\|K(u^{n+1} - \hat{u})\|^2 + \|K(u^{n+1} - u^n)\|^2 - \|K(u^n - \hat{u})\|^2 \\ \quad -\|\hat{z} - z^n - K(\hat{u} - u^{n+1})\|^2 + \|z^n - \hat{z}\|^2 \\ \quad +\|w^n - \hat{w}\|^2 - \|w^{n+1} - w^n\|^2 + \|A^T(w^{n+1} - \hat{w})\|^2 \\ \quad +\|A^T(w^{n+1} - w^n)\|^2 - \|A^T(w^n - \hat{w})\|^2 \\ \quad -\|z^{n+1} - \hat{z}\|^2 - \theta^{-2}\|v^{n+1} - v^n\|^2 + \theta^{-1}\|v^n - \hat{v}\|^2 + \theta^{-1}\|v^{n+1} - v^n\|^2 \end{array} \tag{42}$$

as all remaining inner products cancel. As $\|K\| < 1$ and $\|A\| < 1$ we can introduce real regular square matrices L and B such that $L^T L = 1 - K^T K$ and $B^T B = 1 - AA^T$. The last inequality becomes:

$$\begin{array}{l} \|L(u^{n+1} - \hat{u})\|^2 + \|B(w^{n+1} - \hat{w})\|^2 + \theta^{-1}\|v^{n+1} - \hat{v}\|^2 + \|z^{n+1} - \hat{z}\|^2 \\ \quad \leq \|L(u^n - \hat{u})\|^2 + \|B(w^n - \hat{w})\|^2 + \theta^{-1}\|v^n - \hat{v}\|^2 + \|z^n - \hat{z}\|^2 \\ \quad\quad -\|L(u^{n+1} - u^n)\|^2 - \|B(w^{n+1} - w^n)\|^2 - \|\hat{z} - z^n - K(\hat{u} - u^{n+1})\|^2 \end{array} \tag{43}$$

where we have used that $-\theta^{-2}\|v^{n+1}-v^n\|^2+\theta^{-1}\|v^{n+1}-v^n\|^2 \leq 0$ (for $0<\theta\leq 1$).

It follows from inequality (43) that $(u^n, w^n, v^n, z^n)_n$ is a bounded sequence and that a convergent subsequence $(u^{n_j}, w^{n_j}, v^{n_j}, z^{n_j})_j$ exists. We call the limit of this subsequence $(u^\dagger, w^\dagger, v^\dagger, z^\dagger)$.

Summing inequality (43) from M to $N-1$ yields:

$$\begin{aligned}
&\|L(u^N-\hat{u})\|^2+\|B(w^N-\hat{w})\|^2+\theta^{-1}\|v^N-\hat{v}\|^2+\|z^N-\hat{z}\|^2\\
&\quad\leq \|L(u^M-\hat{u})\|^2+\|B(w^M-\hat{w})\|^2+\theta^{-1}\|v^M-\hat{v}\|^2+\|z^M-\hat{z}\|^2\\
&\qquad-\sum_{n=M}^{N-1}\left(\|L(u^{n+1}-u^n)\|^2+\|B(w^{n+1}-w^n)\|^2+\|\hat{z}-z^n-K(\hat{u}-u^{n+1})\|^2\right).
\end{aligned} \tag{44}$$

It follows that the sum in the right hand side of this expression bounded (independent of N) and therefore that $\|L(u^{n+1}-u^n)\|$, $\|B(w^{n+1}-w^n)\|$ and $\|\hat{z}-z^n-K(\hat{u}-u^{n+1})\|$ tend to zero when n tends to infinity. As B and L are regular, this implies that $\|u^{n+1}-u^n\|$ and $\|w^{n+1}-w^n\|$ tend to zero for large n. Then also $\|\hat{z}-z^n-K(\hat{u}-u^n)\|$ tends to zero for large n. The iteration (34) and relations (36) imply that $v^{n+1}=v^n+\theta(K(u^{n+1}-\hat{u})-(z^{n+1}-\hat{z}))$, and we therefore find that $\|v^{n+1}-v^n\| \overset{n\to\infty}{\longrightarrow} 0$. It follows also from the iteration (34) and the previous remarks that:

$$\begin{aligned}
\|z^{n+1}-z^n\| &= \|Q_{y,\epsilon}(Ku^{n+1}+v^n)-Q_{y,\epsilon}(Ku^n+v^{n-1})\|\\
&\leq \|(Ku^{n+1}+v^n)-(Ku^n+v^{n-1})\|\\
&\leq \|K(u^{n+1}-u^n)\|+\|v^n-v^{n-1}\| \overset{n\to\infty}{\longrightarrow} 0.
\end{aligned}$$

We can therefore conclude that $(u^{n_j+1}, w^{n_j+1}, v^{n_j+1}, z^{n_j+1}) \overset{j\to\infty}{\longrightarrow} (u^\dagger, w^\dagger, v^\dagger, z^\dagger)$ as well. This implies then that $(u^\dagger, w^\dagger, v^\dagger, z^\dagger)$ is a fixed point of the iteration (34) and satisfies the equations (36).

We now choose $(\hat{u}, \hat{w}, \hat{v}, \hat{z}) = (u^\dagger, w^\dagger, v^\dagger, z^\dagger)$ in inequality (44) and find that:

$$\begin{aligned}
&\|L(u^N-u^\dagger)\|^2+\|B(w^N-w^\dagger)\|^2+\theta^{-1}\|v^N-v^\dagger\|^2+\|z^N-z^\dagger\|^2\\
&\quad\leq \|L(u^M-u^\dagger)\|^2+\|B(w^M-w^\dagger)\|^2+\theta^{-1}\|v^M-v^\dagger\|^2+\|z^M-z^\dagger\|^2
\end{aligned}$$

for $N>M$. As there is a convergent subsequence, the right hand side can be made as small as desired by choice of M, and it follows that the entire sequence (u^n, w^n, v^n, z^n) tends to the fixed point $(u^\dagger, w^\dagger, v^\dagger, z^\dagger)$ for large n. □

One can adapt the algorithm (19) (or (34)) and the proof to suit the problem $\min_x H(Ax)+J(Kx)$ (with H and J two convex functions). In this case the projection operator P_μ must be replaced by the proximity operator prox_{H^*} and the projection operator $Q_{y,\epsilon}$ must be replaced by the proximity operator of J: prox_J. As long as these (non-linear) operators have simple expressions, the above algorithms are useable in those cases as well. See [Combettes and Pesquet, 2011] for a general review of proximity operators and [Loris and Verhoeven, 2011] on how proximity operators were used for a generalization of problem (2) and algorithm (11).

6 Conclusions

A set of simple iterative algorithms for the minimization of a penalized least squares functional (2) was presented and their convergence speed was compared numerically in a seismic tomography context ($K \neq \mathrm{Id}$). For the examples considered, the comparison shows that all of these algorithm can produce the minimizer of the functional to within 1 to 10% after about 1000 iterations. The 'best' algorithm depends on the step sizes, the penalty parameter λ and on the metric chosen (functional or distance to minimizer). They can therefore all be used in practice for solving these kind of problems as they appear in seismic tomography.

The four algorithms for minimizing the penalized functional (2) are not new. Here they are presented in a uniform notation making comparison and implementation easier. We also

presented and compared two iterative algorithms for the (equivalent) constrained formulation (3) of this problem. One of these algorithms is new and a proof of convergence is included.

The advantage of the constrained formulation (3) over the penalized formulation (2) is that often the penalty parameter λ has to be determined by trial and error so as to fit the data to the desired level $\|K\hat{u}(\lambda) - y\| \leq \epsilon$. This is done automatically in formulation (3). We have listed simple iterative algorithms for both approaches. However, if the operator A is invertible (and the inverse is readily available), then it can be more advantageous to make a change of variables in the *penalized* formulation and to use an accelerated algorithm (see discussion at end of Section 3.1).

For the readers' convenience we wrote down the explicit forms of several non-smooth convex penalties that can be used for edge-preserving regularization of a seismic imaging problem (total variation penalties and various generalizations) and that can be solved with these algorithms. We also wrote explicit formulas for the convex projection operators that are used by the iterative algorithms in these cases.

We discussed the qualitative properties of these reconstructions on a synthetic seismic tomography example. Particular emphasis was given to the role of sparse local differences. For the case of image denoising $K = \mathrm{Id}$ with TV-like penalties one may refer to [Strong and Chan, 2003] and [Setzer et al., 2011]. Still in the case of denoising, other penalties on local differences (other than ℓ_1-norm type, including non-convex ones), with the goal of maintaining edges in the reconstruction, are compared numerically in [Lukic et al., 2011].

Acknowledgements

The authors would like to acknowledge fruitful discussions with F. Simons, K. Sigloch and G. Nolet. I.L. is a research associate of the Fonds de la recherche Scientifique-FNRS. Part of this research was started while the authors were at CAMP department of the Vrije Universiteit Brussel and was supported by Vrije Universiteit Brussel grant GOA-062 and by the Fonds voor Wetenschappelijk Onderzoek-Vlaanderen grant G.0564.09N.

References

A. Beck and M. Teboulle. A fast iterative shrinkage-threshold algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009. doi: 10.1137/080716542.

K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3:492–526, 2010. doi: 10.1137/090769521.

E. Candes, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Model. Simul.*, 5(3):861–899, Jan. 2006. doi: 10.1137/05064182X.

A. Chambolle. Total variation minimization and a class of binary MRF models. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *Lecture Notes in Computer Science*, pages 136–152. Springer-Verlag Berlin Heidelberg, 2005.

A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188, 1997. doi: 10.1007/s002110050258.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40:120–145, 2011. doi: 10.1007/s10851-010-0251-1.

T. F. Chan, A. Marquina, and P. Mulet. Higher order total variation-based image restoration. *SIAM J. Sci. Comput.*, 22:503–516, 2000. doi: 10.1137/S1064827598344169.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998. doi: 10.1137/S1064827596304010.

A. Cohen, I. Daubechies, and J. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. Pure Appl. Math.*, 45:485–560, 1992. doi: 10.1002/cpa.3160450502.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, 2011.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications On Pure And Applied Mathematics*, 57 (11):1413–1457, Nov. 2004. doi: 10.1002/cpa.20042.

E. Esser, X. Zhang, and T. F. Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM J. Imaging Sci.*, 3(4):1015–1046, Jan. 2010. doi: 10.1137/09076934X.

A. Gholami and H. R. Siahkoohi. Regularization of linear and non-linear geophysical ill-posed problems with joint sparsity constraints. *Geophysical Journal International*, 180(2):871–882, 2010. doi: 10.1111/j.1365-246X.2009.04453.x.

T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2:323–343, 2009. doi: 10.1137/080725891.

G. Hennenfent, E. van den Berg, M. P. Friedlander, and F. J. Herrmann. New insights into one-norm solvers from the Pareto curve. *Geophysics*, 73(4):A23–A26, 2008. doi: 10.1190/1.2944169.

F. J. Herrmann and G. Hennenfent. Non-parametric seismic data recovery with curvelet frames. *Geophysical Journal International*, 173(1):233–248, 2008. doi: 10.1111/j.1365-246X.2007.03698.x.

P. J. Huber. Robust estimation of a location parameter. *Annals of Statistics*, 53:73–101, 1964. doi: 10.1214/aoms/1177703732.

D. Labate, W. Lim, G. Kutyniok, and G. Weiss. Sparse multidimensional representation using shearlets. In *Wavelets XI (San Diego, CA, 2005)*, pages 254–262. SPIE Proc. 5914, SPIE, Bellingham, WA, 2005. doi: 10.1117/12.613494.

I. Loris and C. Verhoeven. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 27:125007, 2011. doi: 10.1088/0266-5611/27/12/125007.

I. Loris, G. Nolet, I. Daubechies, and F. A. Dahlen. Tomographic inversion using ℓ_1-norm regularization of wavelet coefficients. *Geophysical Journal International*, 170(1):359–370, 2007. doi: 10.1111/j.1365-246X.2007.03409.x.

T. Lukic, J. Lindblad, and N. Sladoje. Regularized image denoising based on spectral gradient optimization. *Inverse Problems*, 27(8):085010, 2011. doi: 10.1088/0266-5611/27/8/085010.

S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way.* Academic Press, third edition edition, 2009.

Y. E. Nesterov. A method for solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. *Soviet Math. Dokl.*, 27(2):372–376, 1983.

C. Ronchi, R. Iacono, and P. Paolucci. The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics*, 124: 93–114, 1996. doi: 10.1006/jcph.1996.0047.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, Nov. 1992. doi: 10.1016/0167-2789(92) 90242-F.

S. Setzer, G. Steidl, and T. Teuber. Infimal convolution regularizations with discrete ℓ_1-type functionals. *Communications in Mathematical Sciences*, 9(3):797–827, 2011.

F. J. Simons, I. Loris, G. Nolet, I. C. Daubechies, S. Voronin, J. S. Judd, P. A. Vetter, J. Charléty, and C. Vonesch. Solving or resolving global tomographic models with spherical wavelets, and the scale and sparsity of seismic heterogeneity. *Geophysical Journal International*, 187(2): 969–988, November 2011. doi: 10.1111/j.1365-246X.2011.05190.x.

D. Strong and T. Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, 19(6):S165, 2003. doi: 10.1088/0266-5611/19/6/059.

J. Trampert and J. H. Woodhouse. Global phase-velocity maps of Love and Rayleigh-waves between 40 and 150 seconds. *Geophysical Journal International*, 122(2):675–690, 1995. doi: 10.1111/j.1365-246X.1995.tb07019.x.

J. Trampert and J. H. Woodhouse. High resolution global phase velocity distributions. *Geophysical Research Letters*, 23(1):21–24, 1996. doi: 10.1029/95GL03391.

J. Trampert and J. H. Woodhouse. Assessment of global phase velocity models. *Geophysical Journal International*, 144(1):165–174, 2001. doi: 10.1046/j.1365-246x.2001.00307.x.

E. van den Berg and M. P. Friedlander. Sparse optimization with least-squares constraints. *SIAM J. Optim.*, 21(4):1201–1229, Oct. 2011. doi: 10.1137/100785028.

M. Yamagishi and I. Yamada. Over-relaxation of the fast iterative shrinkage-thresholding algorithm with variable stepsize. *Inverse Problems*, 27(10):105008, 2011. doi: 10.1088/0266-5611/ 27/10/105008.

X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *J. Sci. Comput.*, 46:20–46, 2011. doi: 10.1007/s10915-010-9408-8.